

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ



ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΜΕΘΟΔΟΛΟΓΙΑ COLUMN GENERATION ΓΙΑ ΒΕΛΤΙΣΤΗ
ΑΝΑΘΕΣΗ ΙΠΤΑΜΕΝΟΥ ΠΡΟΣΩΠΙΚΟΥ
(COLUMN GENERATION METHODOLOGY FOR OPTIMAL
CREW SCHEDULING)**

υπό τους

ΖΟΡΜΠΑ ΔΗΜΗΤΡΙΟ

ΨΑΡΟΓΙΩΡΓΟ ΣΠΥΡΙΔΩΝ

ΑΠΟΦΟΙΤΟΥΣ ΣΧΟΛΗΣ ΙΚΑΡΩΝ (ΣΜΑ), 2013

Υπεβλήθη για την εκπλήρωση μέρους των
απαιτήσεων για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης

ΒΟΛΟΣ 2016

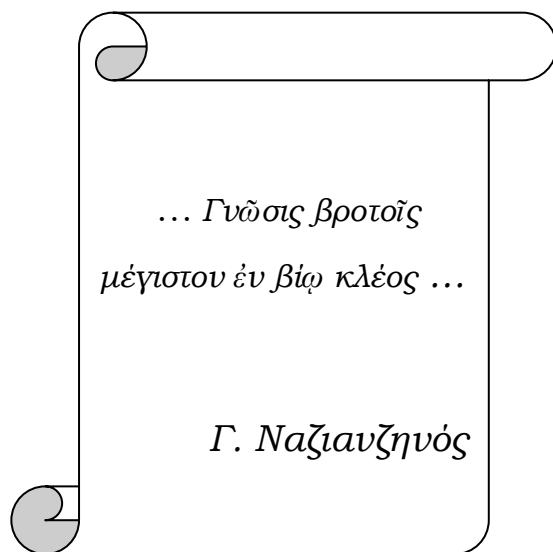
© 2016 Ζορμπάς Δημήτριος, Ψαρογιώργος Σπυρίδων

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).



*... Γνώσις βροτοῖς
μέγιστον ἐν βίῳ κλέος ...*

Γ. Ναζιανζηνός

Αφιερώνεται ...

*Στις οικογένειές μας που με τη βοήθεια,
την υποστήριξη και την υπομονή τους συνέβαλαν στην ολοκλήρωση
των μεταπτυχιακών σπουδών μας.*

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων) Δρ. Κοζανίδης Γεώργιος
Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής Δρ. Λυμπερόπουλος Γεώργιος
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο
Θεσσαλίας

Τρίτος Εξεταστής Δρ. Ζηλιασκόπουλος Αθανάσιος
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο
Θεσσαλίας

Ευχαριστίες

Πρώτα απ' όλα, θέλουμε να ευχαριστήσουμε τον επιβλέποντα της μεταπτυχιακής εργασίας μας, Επίκουρο Καθηγητή κ. Γεώργιο Κοζανίδη, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μας. Επίσης, είμαστε ευγνώμονες στα υπόλοιπα μέλη της εξεταστικής επιτροπής της μεταπτυχιακής εργασίας μας, Καθηγητές κ. Γεώργιο Λυμπερόπουλο και κ. Αθανάσιο Ζηλιασκόπουλο, για την προσεκτική ανάγνωση της εργασίας μας και για τις πολύτιμες υποδείξεις τους. Πάνω απ' όλα, είμαστε ευγνώμονες στους γονείς μας, για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια, στους οποίους και αφιερώνουμε αυτήν την μεταπτυχιακή εργασία.

ΖΟΡΜΠΙΑΣ ΔΗΜΗΤΡΙΟΣ

και

ΨΑΡΟΓΙΩΡΓΟΣ ΣΠΥΡΙΔΩΝ

**ΜΕΘΟΔΟΛΟΓΙΑ COLUMN GENERATION ΓΙΑ ΒΕΛΤΙΣΤΗ
ΑΝΑΘΕΣΗ ΙΠΤΑΜΕΝΟΥ ΠΡΟΣΩΠΙΚΟΥ
(COLUMN GENERATION METHODOLOGY FOR OPTIMAL
CREW SCHEDULING)**

ΖΟΡΜΠΙΑΣ ΔΗΜΗΤΡΙΟΣ

ΨΑΡΟΓΙΩΡΓΟΣ ΣΠΥΡΙΔΩΝ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, 2016

Επιβλέπων Καθηγητής: Δρ. Κοζανίδης Γεώργιος, Επίκουρος Καθηγητής
Επιχειρησιακής Έρευνας

ΠΕΡΙΛΗΨΗ

Ο προγραμματισμός πληρώματος αποτελεί μία διαδικασία που ακολουθείται από τις αεροπορικές εταιρίες με σκοπό την εύρεση του καλύτερου τρόπου διαχείρισης του πληρώματος και των αεροσκαφών τους. Το πρόβλημα του προγραμματισμού πληρώματος παρουσιάζει ένα σημαντικό βαθμό πολυπλοκότητας με κύριο στόχο την ελαχιστοποίηση του κόστους με το οποίο επιβαρύνεται η αεροπορική εταιρία. Σε αυτή τη διπλωματική εργασία αναπτύσσουμε ένα αλγόριθμο για την επίλυση του προβλήματος της κάλυψης ενός συνόλου διαδρομών/πτήσεων από ένα συγκεκριμένο αριθμό πιλότων, ικανοποιώντας ταυτόχρονα ένα σύνολο περιορισμών.

Αρχικά, περιγράφουμε τα βασικά προβλήματα που έχουν να αντιμετωπίσουν οι αεροπορικές εταιρίες όπως είναι ο προγραμματισμός των πτήσεων, ο σχεδιασμός του δικτύου πτήσεων, η ανάθεση του πληρώματος, η δημιουργία περιόδων υπηρεσίας καθώς και οι δομές κόστους και αμοιβών.

Στη συνέχεια, γίνεται μία βιβλιογραφική ανασκόπηση των παραπάνω προβλημάτων καθώς και των τεχνικών που έχουν προταθεί για την επίλυσή τους. Οι τεχνικές αυτές είναι κυρίως μαθηματικές και αντιμετωπίζουν το πρόβλημα του προγραμματισμού πληρώματος ως ένα μοντέλο ακέραιου προγραμματισμού.

Η μαθηματική μοντελοποίηση του προγραμματισμού πληρώματος γίνεται έπειτα από την ανάλυση του προβλήματος, των περιορισμών που το διέπουν καθώς και του αντικειμενικού σκοπού του. Η μέθοδος που ακολουθείται είναι η branch and price που αποτελείται από το master problem και το column generation sub-problem. Έτσι, γίνεται η περιγραφή των εξισώσεων με σκοπό τη μαθηματική μοντελοποίηση του προβλήματος ώστε να εξαχθεί ο αλγόριθμος επίλυσης.

Η μεθοδολογία επίλυσης που ακολουθεί την παραπάνω μοντελοποίηση αποτελείται από την αρχικοποίηση του restricted master problem (RMP), την επίλυση του master LP relaxation, την επίλυση του column generation sub-problem και τέλος την εφαρμογή του βήματος braching. Για όλα τα παραπάνω βήματα είναι

απαραίτητος ο ορισμός των παραμέτρων, των εξισώσεων καθώς και των περιορισμών και επιλύονται με τη βοήθεια του κατάλληλου λογισμικού.

Τέλος, επιλύεται μια σειρά από προβλήματα με σκοπό την παρουσίαση της λειτουργίας του αλγορίθμου, την εξέταση της συμπεριφοράς του, καθώς και την εξαγωγή των σχετικών συμπερασμάτων. Η παρούσα διπλωματική εργασία κλείνει με την παρουσίαση των γενικών συμπερασμάτων και τις προτάσεις για μελλοντική μελέτη.

ABSTRACT

Crew scheduling is a procedure that has been widely followed by airlines in order to find the most efficient way to manage their crew and fleet. The problem of crew scheduling is characterized by a high level of complexity. The main target is to find the minimum cost for the airline. In this thesis, we develop an algorithm in order to cover a set of routes/flights with a given number of pilots, while also satisfying at the same time a given number of restrictions.

Firstly, we describe typical problems that airlines are faced with, such as flight scheduling, the design of operation network, crew pairing, duty periods and payment structures. Then, we present a literature review on the above problems and the various techniques that have been developed for their treatment. These are mathematical programming techniques, which address the problem of crew scheduling as an integer programming problem. The most popular solution approaches for such problems are branch and bound, branch and price and column generation.

After the definition of the problem, the constraints and the main objectives, the crew scheduling optimization model is developed. The main solution technique that we develop is branch and price which utilizes a master problem and a column generation sub-problem. Thus, the equations that are developed for modeling the problem are described in order to derive the proposed algorithm.

The proposed solution methodology involves the initialization of the restricted master problem (RMP), the solution of the master LP relaxation, the solution of the column generation sub-problem and the application of branching step. For all these steps, we need to define the parameters, the equations and the restrictions and use the appropriate software.

Finally, a number of problems are being solved in order to demonstrate the performance of the algorithm that we have developed and summarize our conclusions. The present thesis closes with the presentation of general conclusions and suggestions for future study.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1	16
ΕΙΣΑΓΩΓΗ.....	16
1.1 Γενική Περιγραφή του Προβλήματος.....	16
1.2 Προγραμματισμός Πληρώματος (Crew Scheduling).....	19
1.3 Προγραμματισμός Αεροπορικών Γραμμών.....	20
1.4 Το Πρόβλημα του Προγραμματισμού Πληρώματος	22
1.4.1 Κανόνες Εργασίας και Δομή Αμοιβών	23
1.4.1.1 Περίοδος Υπηρεσιών (Duty Periods)	24
1.4.1.2 Ζεύγη (Pairings).....	24
1.4.1.3 Πρόγραμμα (Schedule)	25
1.4.2 Το Πρόβλημα των Ζευγών και της Εκχώρησης Πληρώματος	26
1.4.2.1 Το Πρόβλημα των Ζευγών.....	26
1.4.2.2 Το Πρόβλημα Εκχώρησης Πληρώματος	28
ΚΕΦΑΛΑΙΟ 2	31
ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....	31
2.1 Εισαγωγή.....	31
2.2 Ανάθεση Στόλου (Fleet Assignment)	31
2.3 Δρομολόγηση αεροσκαφών (Aircraft Routing).....	32
2.4 Προγραμματισμός Πληρώματος (Crew Scheduling).....	33
2.4.1 Παράλληλες Προσεγγίσεις στο Πρόβλημα Ταιριάσματος Πληρώματος (Crew Pairing).....	36
2.4.2 Παράλληλες Προσεγγίσεις στην Ανάθεση Πληρώματος (Crew Rostering).....	37

2.5 Μαθηματικές τεχνικές προγραμματισμού	38
2.5.1 Branch-and-Bound	38
2.5.2 Branch and Price	39
2.5.3 Column Generation	41
ΚΕΦΑΛΑΙΟ 3	42
ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ	42
3.1 Εισαγωγή.....	42
3.2 Master Problem	42
3.3 Column Generation Sub-problem	48
3.4 Branching	52
ΚΕΦΑΛΑΙΟ 4	55
ΜΕΘΟΔΟΛΟΓΙΑ ΕΠΙΛΥΣΗΣ	55
4.1 Εισαγωγή.....	55
4.2 Αρχικοποίηση του RMP	55
4.3 Επίλυση του master LP relaxation	57
4.4 Αριθμός μεταβλητών που εισάγονται ανά επανάληψη.....	59
4.5 Επίλυση του column generation sub-problem	60
4.6 Εφαρμογή του βήματος Branching	64
4.7 Επίλυση αριθμητικού παραδείγματος.....	66
ΚΕΦΑΛΑΙΟ 5	85
ΥΠΟΛΟΓΙΣΤΙΚΗ ΕΜΠΕΙΡΙΑ.....	85
5.1 Εισαγωγή.....	85
5.2 Εφαρμογές του αλγορίθμου σε προβλήματα διαφορετικού μεγέθους.....	88
5.2.1 Εφαρμογή 1 ^η	88
5.2.2 Εφαρμογή 2 ^η	89

5.2.3	Εφαρμογή 3 ^η	90
5.2.4	Εφαρμογή 4 ^η	92
5.2.5	Εφαρμογή 5 ^η	93
5.2.6	Εφαρμογή 6 ^η	94
5.2.6.1	Τιμή παραμέτρου paths=2.....	94
5.2.6.2	Τιμή παραμέτρου paths=3.....	95
5.2.6.3	Τιμή παραμέτρου paths=5.....	96
5.2.6.4	Τιμή παραμέτρου paths=10.....	97
ΚΕΦΑΛΑΙΟ 6		98
ΣΥΝΟΨΗ, ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ		98
6.1	Σύνοψη.....	98
6.2	Συμπεράσματα	99
6.3	Μελλοντική Έρευνα.....	105
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		107
ΠΑΡΑΡΤΗΜΑ 1		111
ΠΑΡΑΡΤΗΜΑ 2		118

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

<i>Εικόνα 2.1: Ένα τυπικό ζεύγος μέσα σε μια περίοδο υπηρεσίας</i>	<i>35</i>
<i>Εικόνα 4.1: Στοιχεία προβλήματος.</i>	<i>67</i>
<i>Εικόνα 4.2: Δημιουργία μεταβλητών για το RMP</i>	<i>68</i>
<i>Εικόνα 4.3: Δημιουργία περιορισμών του RMP.....</i>	<i>69</i>
<i>Εικόνα 4.4: Εξισώσεις του RMP.....</i>	<i>71</i>
<i>Εικόνα 4.5 Εξισώσεις RMP μετά την 1^η επανάληψη του column generation.</i>	<i>75</i>
<i>Εικόνα 4.6: Διαγραφή μεταβλητών από τον περιορισμό του πιλότου.</i>	<i>79</i>
<i>Εικόνα 4.7: Διαγραφή μεταβλητών από περιορισμό διαδρομής.</i>	<i>79</i>
<i>Εικόνα 4.8: Τελική λύση προβλήματος.....</i>	<i>81</i>
<i>Εικόνα 4.9: Τελικές εξισώσεις του προβλήματος.</i>	<i>84</i>

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

<i>Σχήμα 1.1: Βήματα επίλυσης προβλημάτων αεροπορικών εταιριών.....</i>	<i>21</i>
<i>Σχήμα 1.2 : Διάγραμμα του προβλήματος ανάθεσης πληρώματος σε πτήσεις</i>	<i>30</i>
<i>Σχήμα 4.1: Βήματα για την επίλυση του προβλήματος.</i>	<i>58</i>
<i>Σχήμα 4.2: Αρχικοποίηση των τιμών των κόμβων (πάνω είναι η αρχικοποίηση για έναν συμβατό κόμβο, κάτω ο κόμβος j' δεν είναι συμβατός).</i>	<i>61</i>
<i>Σχήμα 4.3: Ετικέτες των κόμβων πριν τον έλεγχο για συνδυασμό.....</i>	<i>62</i>
<i>Σχήμα 4.4: Τιμές κόμβου t μετά τον έλεγχο.....</i>	<i>63</i>
<i>Σχήμα 4.5: Δέντρο διακλάδωσης (Branching tree).....</i>	<i>65</i>
<i>Σχήμα 4.6: Branching tree.</i>	<i>80</i>
<i>Σχήμα 6.1: Ελάχιστη, μέση και μέγιστη τιμή της τελικής βέλτιστης λύσης των εφαρμογών, όπου $paths=1$.</i>	<i>100</i>
<i>Σχήμα 6.2: Ελάχιστη, μέση και μέγιστη τιμή της τελικής βέλτιστης λύσης για την εφαρμογής 6, για διάφορες τιμές της παραμέτρου $paths$.....</i>	<i>101</i>
<i>Σχήμα 6.3: Ποσοστό εμφάνισης αρνητικού gap για τις διάφορες τιμές της παραμέτρου $paths$.....</i>	<i>102</i>
<i>Σχήμα 6.4: Επίδραση του μεγέθους του προβλήματος στον χρόνο επίλυσής του.</i>	<i>103</i>
<i>Σχήμα 6.5: Επίδραση της παραμέτρου $paths$ στον χρόνο επίλυσης.</i>	<i>104</i>

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

<i>Πίνακας 5.1: Συγκεντρωτικός πίνακας αποτελεσμάτων.....</i>	<i>87</i>
<i>Πίνακας 5.2: Αποτελέσματα 1^{ης} εφαρμογής.....</i>	<i>88</i>
<i>Πίνακας 5.3: Αποτελέσματα 2^{ης} εφαρμογής.....</i>	<i>89</i>
<i>Πίνακας 5.4: Αποτελέσματα 3^{ης} εφαρμογής.....</i>	<i>91</i>
<i>Πίνακας 5.5: Αποτελέσματα 4^{ης} εφαρμογής.....</i>	<i>92</i>
<i>Πίνακας 5.6: Αποτελέσματα 5^{ης} εφαρμογής.....</i>	<i>93</i>
<i>Πίνακας 5.7: Αποτελέσματα 6^{ης} εφαρμογής για paths=2.</i>	<i>94</i>
<i>Πίνακας 5.8: Αποτελέσματα 6^{ης} εφαρμογής για paths=3.</i>	<i>95</i>
<i>Πίνακας 5.9: Αποτελέσματα 6^{ης} εφαρμογής για paths=5.</i>	<i>96</i>
<i>Πίνακας 5.10: Αποτελέσματα 6^{ης} εφαρμογής για paths=10.</i>	<i>97</i>

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

TAFB (Time Away From Base)

ΟΔΑ (Ομοσπονδιακή Διοίκηση Αεροπορίας)

RMP (Restricted Master Problem)

MP (Master Problem)

ASPA (Acyclic Shortest Path Algorithm)

LP (Linear Programming)

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Γενική Περιγραφή του Προβλήματος

Η αποτελεσματική λειτουργία μιας αεροπορικής εταιρείας απαιτεί τον κατάλληλο προγραμματισμό και τις απαραίτητες μεθοδολογίες για την διεκπεραίωση των ιδιαίτερα πολύπλοκων και πολυδιάστατων δραστηριοτήτων της. Καθώς οι οικονομικές αγορές εξελίσσονται και γίνονται παγκόσμιες, οι αεροπορικές εταιρείες οδηγούνται στην επιτακτική ανάγκη για κατάκτηση όλο και μεγαλύτερου ανταγωνιστικού μεριδίου της παγκόσμιας αγοράς, με αποτέλεσμα να στρέφονται σε όλο και περισσότερες και πιο ανεπτυγμένες τεχνικές βελτιστοποίησης. Αυτό έχει ως απώτερο σκοπό την δημιουργία κατάλληλων συστημάτων υποστήριξης και λήψης αποφάσεων, τα οποία έχουν ως στόχο τον καλύτερο έλεγχο και την καλύτερη λειτουργία της αεροπορικής εταιρείας, αλλά και όλων των διαδικασιών της.

Όπως γίνεται εύκολα αντιληπτό, όλα τα παραπάνω οδήγησαν στην αύξηση της πολυπλοκότητας των λειτουργιών των αεροπορικών εταιρειών, τόσο σε επιχειρησιακό όσο και σε τακτικό επίπεδο.. Οι σημερινές αεροπορικές εταιρείες καλούνται να βρουν τον καλύτερο τρόπο διαχείρισης του πληρώματος και των αεροσκαφών τους, με στόχο την βελτιστοποίηση της λειτουργίας των πτήσεων, αλλά και κυρίως την ελαχιστοποίηση του κόστους λειτουργίας τους. Για να ανταπεξέλθουν επιτυχώς στις απαιτήσεις και να προσφέρουν ένα υψηλής ποιότητας, αλλά και χαμηλού κόστους προϊόν, οι αεροπορικές εταιρείες επενδύουν τεράστια υλικά αλλά και πνευματικά αποθέματα στην προσπάθειά τους να εξελίξουν οικονομικά αποδοτικές μεθόδους αντιμετώπισης των κυριότερων προβλημάτων τους. Τα προβλήματα, τα οποία έχουν να αντιμετωπίσουν οι αεροπορικές εταιρείες είναι κατά βάση ο προγραμματισμός των πτήσεων, ο σχεδιασμός δικτύου πτήσεων,

η ακολουθία των πτήσεων, η ανάθεση πληρωμάτων, η ανάθεση πυλών αναχώρησης και άφιξης ανά αεροδρόμιο, το πρόγραμμα συντήρησης αεροσκαφών, το πρόγραμμα εκπαίδευσης και ξεκούρασης του πληρώματος, καθώς και διαδικασίες που αφορούν την τιμολόγηση των εισιτηρίων και τη διαχείριση των αποσκευών.

Όλα τα παραπάνω προβλήματα έχουν ένα σημαντικό βαθμό πολυπλοκότητας, ο οποίος προέρχεται από το ίδιο το περιβάλλον, στο οποίο λειτουργούν οι αεροπορικές εταιρείες. Οι αεροπορικές εταιρείες, κινούνται μέσα σε ένα εξαιρετικά δυναμικό περιβάλλον, το οποίο εναλλάσσεται διαρκώς. Δεν είναι λίγες οι φορές που μια εταιρεία καλείται να τροποποιήσει το αρχικό της πρόγραμμα εξαιτίας έκτακτων γεγονότων, όπως είναι οι καιρικές συνθήκες, οι μηχανικές βλάβες στα αεροσκάφη, η ασθένεια του πληρώματος είτε ακόμα και αλλαγές στις τιμές του πετρελαίου ή μιας απρόσμενης τρομοκρατικής ενέργειας, η οποία μπορεί να επηρεάσει τη ζήτηση στις αερομεταφορές.

Εκτός αυτού, οι αεροπορικές εταιρείες επηρεάζονται και από νομικούς και άλλους παράγοντες, οι οποίοι διέπουν τις εναέριες μεταφορές και συνήθως σχετίζονται με τα μέτρα ασφαλείας, τα οποία είναι αναγκαία να λαμβάνονται από κάθε εταιρεία, με τη συντήρηση του στόλου των αεροσκαφών, με την εκπαίδευση του πληρώματος και με το μέγιστο επιτρεπτό χρόνο πτήσεων του πληρώματος. Τα στενά οικονομικά όρια, τα οποία τίθενται στον γενικότερο προγραμματισμό της εταιρείας, αλλά και η επιτακτική ανάγκη για εκμετάλλευση των διαθέσιμων πόρων, στον μέγιστο δυνατό και επιτρεπτό βαθμό, δεν αφήνουν περιθώρια ελιγμών ή λάθους στον προγραμματισμό των πτήσεων, του στόλου των αεροσκαφών, αλλά και του προσωπικού της επιχείρησης.

Λόγω της πολυπλοκότητας και της δυσκολίας της διαχείρισης συστημάτων τέτοιας μεγάλης κλίμακας, στο πλαίσιο της Επιχειρησιακής Έρευνας και της Διοικητικής Επιστήμης, έχουν αναπτυχθεί πολλά μαθηματικά μοντέλα και ποσοτικές μέθοδοι για την προσομοίωση και τη βελτιστοποίηση της λειτουργίας τους. Τα μοντέλα αυτά, σε συνδυασμό με διάφορα πακέτα λογισμικού, παρέχουν μια καλή υποστήριξη στους εκάστοτε φορείς λήψης αποφάσεων.

Ωστόσο, πολλά από αυτά τα μοντέλα, προκειμένου να γίνουν πιο απλά περιέχουν υποθέσεις, οι οποίες είναι μη ρεαλιστικές για πρακτικές εφαρμογές, καθώς

σε πραγματικά προβλήματα υπάρχουν καταστάσεις, οι οποίες συχνά ξεφεύγουν της μαθηματικής μοντελοποίησης. Ως εκ τούτου, προκειμένου να καλυφθεί η ανάγκη για πιο ευέλικτα και σύγχρονα εργαλεία, η έρευνα επικεντρώθηκε σε πιο εξελιγμένα και πολύπλοκα μοντέλα, τα οποία να μπορούν να διαχειριστούν και πιο μεγάλα προβλήματα.

Τέτοια μεγάλα και σύνθετα προβλήματα, είναι τα προβλήματα βελτιστοποίησης που αντιμετωπίζουν οι αεροπορικές εταιρείες, τα οποία αναφέραμε και πιο πάνω. Τα προβλήματα αυτά για την επίλυσή τους απαιτούν καινοτόμες λύσεις και ιδέες. Έτσι, η ανάπτυξη νέων μοντέλων και αλγορίθμων οδήγησε στην δυνατότητα βελτίωσης της παραγωγικότητας των πόρων, καθώς και της αποτελεσματικότητας και της ικανότητας του δικτύου μεταφοράς. Επομένως, έχει γίνει πλέον αποδεκτό από όλους, ότι η ανάπτυξη ενός συστήματος υποστήριξης αποφάσεων, το οποίο βασίζεται στη μοντελοποίηση, στην ανάπτυξη αλγορίθμων και στη βελτιστοποίηση, αποτελεί αναπόσπαστο κομμάτι της διαδικασίας λήψης αποφάσεων.

Από άποψη μοντελοποίησης πολλά προβλήματα που απασχολούν τις αεροπορικές εταιρείες, μετατρέπονται σε συνδυαστικής βελτιστοποίησης προβλήματα, τα οποία είναι αρκετά δύσκολο να επιλυθούν. Ωστόσο, η εξέλιξη στον τομέα των αλγορίθμων, κυρίως με την εμφάνιση αρκετών ευρετικών αλγορίθμων, αλλά και την τεράστια και συνεχώς αυξανόμενη πρόοδο στον τομέα των ηλεκτρονικών υπολογιστών, καθιστά δυνατή την επίλυση σε πραγματικό χρόνο δύσκολων και μεγάλων προβλημάτων με αρκετά αποτελεσματικό τρόπο.

Η παρούσα διπλωματική εργασία ασχολείται με το πρόβλημα της κάλυψης ενός συνόλου διαδρομών (routes), σε ένα ολοκληρωμένο δίκτυο διανομής, από ένα συγκεκριμένο αριθμό πιλότων, ικανοποιώντας ταυτόχρονα και ένα σύνολο ποιοτικών κριτηρίων, όπως η μη κάλυψη αλληλεπικαλυπτόμενων χρονικά αποστολών από τον ίδιο πιλότο, οι μη διαθέσιμες ημερομηνίες πτήσης κάθε πιλότου κ.α. Η χρονική περίοδος, για την οποία λύνεται το πρόβλημα είναι για διάστημα ενός μήνα.

1.2 Προγραμματισμός Πληρώματος (Crew Scheduling)

Ο προγραμματισμός του πληρώματος (crew scheduling) μπορεί να προσδιορισθεί ως το πρόβλημα της ανάθεσης σε μια ομάδα εργατών/πληρώματος ενός συνόλου εργασιών. Τα πληρώματα μπορούν τυπικά να ανταλλάσσονται αν και σε μερικές περιπτώσεις διαφορετικά πληρώματα κατέχουν διαφορετικά χαρακτηριστικά τα οποία επηρεάζουν ποια σύνολα εργασιών μπορούν να ολοκληρώσουν.

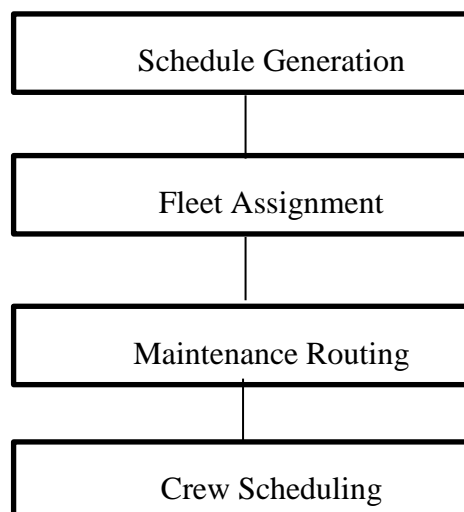
Τα προβλήματα του προγραμματισμού του πληρώματος εμφανίζονται σε μεγάλο αριθμό μεταφορικών μέσων, για παράδειγμα στα λεωφορεία, στα τρένα, στις μεταφορές εμπορευμάτων με φορτηγά και αμαξοστοιχίες και στις αεροπορικές μεταφορές. Υπάρχουν πολλά κοινά στοιχεία σε όλα αυτά τα προβλήματα στα οποία συμπεριλαμβάνεται η ανάγκη κάλυψης όλων των εργασιών με την ταυτόχρονη ελαχιστοποίηση του κόστους εργασίας και η εισαγωγή ενός πλήθους περιορισμών οι οποίοι υπαγορεύονται από τους κανονισμούς ασφαλείας και τις εργασιακές διαπραγματεύσεις. Παρ' όλα αυτά, η κάθε περίπτωση έχει τα δικά της χαρακτηριστικά και τις δικές της ερευνητικές προκλήσεις. Στην πραγματικότητα, οι περισσότερες έρευνες πάνω στον προγραμματισμό πληρώματος εστιάζουν σε συγκεκριμένες εφαρμογές παρά σε γενικές περιπτώσεις.

Ο προγραμματισμός πληρώματος στις αεροπορικές εταιρείες παρουσιάζει ιδιαίτερο ενδιαφέρον για διάφορους λόγους. Πρώτον, παρέχει ένα πλαίσιο για την εξέταση πολλών στοιχείων που είναι παρόμοια με όλα τα προβλήματα που παρουσιάζονται κατά τον προγραμματισμό του πληρώματος. Δεύτερον, ο προγραμματισμός στις αερογραμμές αποτελεί ένα πραγματικό πρόβλημα προγραμματισμού με την έννοια ότι οι αεροπορικές εταιρείες έχουν σταθερό πρόγραμμα το οποίο αλλάζει το πολύ μηνιαία και έτσι υπάρχει ουσιαστικός χρόνος και πόροι που μπορούν να χρησιμοποιηθούν για την επίλυσή του. Τρίτον, το προσωπικό των αεροπορικών εταιριών λαμβάνει υψηλότερο μισθό από αντίστοιχο προσωπικό σε άλλο μεταφορικό μέσο και με αυτόν τον τρόπο τα κέρδη που θα υπάρξουν από ένα βελτιωμένο προγραμματισμό πληρώματος θα είναι σημαντικά. Τέλος, ένας μεγάλος αριθμός αυστηρών κανονισμών που διέπουν τις αερομεταφορές

και οι ισχυρές εργατικές ενώσεις περιορίζουν σε μεγάλο βαθμό το σύνολο των εφικτών λύσεων κάνοντας τον προγραμματισμό πληρώματος στις αεροπορικές γραμμές ένα από τα δυσκολότερα προβλήματα προγραμματισμού πληρώματος. Για όλους αυτούς τους λόγους, ο προγραμματισμός του πληρώματος στις αερογραμμές έχει λάβει την μεγαλύτερη προσοχή τόσο από την βιομηχανία όσο και από την ακαδημαϊκή κοινότητα.

1.3 Προγραμματισμός Αεροπορικών Γραμμών

Ο προγραμματισμός του πληρώματος είναι ένα μόνο μέρος των προβλημάτων που αντιμετωπίζουν οι αεροπορικές γραμμές. Παρά το γεγονός ότι αυτά τα προβλήματα σχετίζονται μεταξύ τους, επιλύονται διαδοχικά ανάλογα με το μέγεθος και την πολυπλοκότητά τους. Οι αεροπορικές εταιρείες ξεκινούν συνήθως λύνοντας το πρόβλημα σχεδιασμού των δρομολογίων, στο οποίο καθορίζουν τις πτήσεις που θα πραγματοποιηθούν σε μια συγκεκριμένη περίοδο. Στο επόμενο βήμα βρίσκεται το πρόβλημα της ανάθεσης στόλου κατά το οποίο αποφασίζουν το είδος του αεροσκάφους (για παράδειγμα Boeing 767, 727, 777 κλπ) που θα εκχωρηθεί σε κάθε πτήση συναρτήσει της ζήτησης που προβλέπεται να έχει. Ακολουθεί το πρόβλημα της συντήρησης κατά τα δρομολόγια στο οποίο πρέπει να ανατεθεί στο κάθε αεροσκάφος πτήση η οποία να εξασφαλίζει επαρκή χρόνο σε συγκεκριμένα αεροδρόμια στα οποία θα περάσει από ελέγχους ρουτίνας για τη συντήρησή του. Ολοκληρώνοντας αυτά τα τρία στάδια, οι αερογραμμές δρομολογούν το πρόβλημα του προγραμματισμού του πληρώματος.



Σχήμα 1.1: Βήματα επίλυσης προβλημάτων αεροπορικών εταιριών

Με τον προγραμματισμό του πλήρωματος στις αερογραμμές υπάρχουν σημαντικές διαφοροποιήσεις στον τρόπο με τον οποίο σχεδιάζονται οι διεθνείς και οι εγχώριες πτήσεις. Στις Η.Π.Α., για παράδειγμα, τα δίκτυα των διεθνών πτήσεων τείνουν να είναι πιο αραιά με περιορισμένο αριθμό πτήσεων από και προς το κάθε αεροδρόμιο ενώ σε αντίθεση, τα δίκτυα των πτήσεων εσωτερικού χαρακτηρίζονται από μεγάλο αριθμό αναχωρήσεων που ακολουθείται από μεγάλο αριθμό αφίξεων στα κεντρικά αεροδρόμια σε σχετικά σύντομο χρονικό διάστημα. Ωστόσο, τα δίκτυα διεθνών πτήσεων χαρακτηρίζονται ως σημείο-σε-σημείο δίκτυα με τις διάφορες λειτουργίες να μοιράζονται μέσα τους. Μια άλλη διαφορά είναι ότι τα διεθνή δρομολόγια συνήθως λειτουργούν με εβδομαδιαίο πρόγραμμα ενώ καθημερινά προγράμματα χρησιμοποιούνται συνήθως σε πτήσεις εσωτερικού. Επιπλέον, σε αντίθεση με τα δρομολόγια εσωτερικού, δεν είναι απίθανο στις διεθνείς πτήσεις η αεροπορική εταιρεία να μεταφέρει το πλήρωμα της ως επιβάτες σε κάποια από τις πτήσεις της με σκοπό την επανατοποθέτησή του σε κάποια μελλοντική πτήση από το αεροδρόμιο άφιξης. Όλες αυτές οι διαφορές επηρεάζουν τον τρόπο με τον οποίο προγραμματίζεται το πλήρωμα.

Υπάρχουν, επίσης, σημαντικές διαφορές μεταξύ του τρόπου με τον οποίο προγραμματίζεται το πλήρωμα στο πιλοτήριο και του τρόπου που προγραμματίζεται το πλήρωμα στην καμπίνα του αεροσκάφους. Για παράδειγμα, οι πιλότοι και το

υπόλοιπο πλήρωμα του πιλοτηρίου συνήθως παραμένουν μαζί στο μεγαλύτερο μέρος του προγράμματός τους. Το πλήρωμα της καμπίνας τείνει να διαφοροποιείται πιο συχνά με τους αεροσυνοδούς να προγραμματίζονται περισσότερο ατομικά παρά σαν μέρος προκαθορισμένης ομάδας. Άλλη μια διαφορά-κλειδί είναι ότι ο πιλότος και το πλήρωμα του πιλοτηρίου περιορίζονται αυστηρά από το είδος του αεροσκάφους για το οποίο έχουν τα προσόντα να πετάξουν ενώ το πλήρωμα της καμπίνας έχει μεγαλύτερο εύρος τύπου αεροσκαφών που μπορεί να εκχωρηθεί. Έτσι, θα επικεντρωθούμε στο πρόβλημα προγραμματισμού του πληρώματος του πιλοτηρίου.

1.4 Το Πρόβλημα του Προγραμματισμού Πληρώματος

Το κάθε μέλος του πληρώματος πιλοτηρίου είναι αρμόδιο να πετάξει ένα συγκεκριμένο είδος αεροσκάφους ή στενά συνδεδεμένα είδη αεροσκαφών, γνωστά ως «οικογένεια αεροσκαφών» (fleet family). Επομένως, επιλύουμε ένα ξεχωριστό πρόβλημα προγραμματισμού πληρώματος για κάθε τύπο αεροσκάφους, το οποίο περιλαμβάνει μόνο τις πτήσεις που έχουν ανατεθεί στο συγκεκριμένο είδος αεροσκάφους.

Τα δεδομένα που εισάγονται στο πρόβλημα του προγραμματισμού είναι το σετ των πτήσεων που πρέπει να καλυφθεί. Οι πτήσεις ομαδοποιούνται προκειμένου να σχηματίσουν περιόδους υπηρεσιών (duty periods) οι οποίες είναι μια σειρά από επακόλουθες πτήσεις που περιλαμβάνουν μια μέρα εργασίας του πληρώματος. Έπειτα, οι υπηρεσίες (duties) σχηματίζουν ζευγάρια ώστε τα ταξίδια του πληρώματος να συνδέουν μία ή περισσότερες μέρες εργασίας τις οποίες χωρίζουν περίοδοι ξεκούρασης. Τέλος, τα μηνιαία προγράμματα δημιουργούνται από πολλαπλά ζεύγη υπηρεσιών με διαλείμματα, δηλαδή χρόνο ξεκούρασης, ενδιάμεσα. Οι πτήσεις, οι υπηρεσίες, τα ζεύγη και τα μηνιαία προγράμματα είναι τα θεμέλια του προγραμματισμού πληρώματος αεροσκαφών.

Με κάθε ένα από αυτά τα θεμέλια συνδέεται ένα σαφές σύνολο περιορισμών. Οι περιορισμοί αυτοί προέρχονται, τυπικά, από τρεις πηγές. Οι πρώτοι περιορισμοί προέρχονται από τους κανονισμούς που επιβάλλουν οι οργανισμοί της πολιτικής αεροπορίας οι οποίοι περιορίζουν τον προγραμματισμό του πληρώματος κυρίως για λόγους ασφαλείας. Δεύτερον, τα εργατικά σωματεία πετυχαίνουν, έπειτα από συλλογικές διαπραγματεύσεις, συμφωνίες που αφορούν τις συνθήκες εργασίας του πληρώματος. Τέλος, η τρίτη πηγή περιορισμών είναι οι ίδιες οι αεροπορικές εταιρείες που πολλές φορές προσθέτουν επιπλέον περιορισμούς με σκοπό να κάνουν πιο εύρωστο το πρόγραμμά τους. Επιπλέον αυτών των περιορισμών, το κάθε θεμέλιο του προγραμματισμού σχετίζεται με μια σαφή δομή κόστους. Έτσι, οι αεροπορικές εταιρείες συναντούν περιορισμούς τόσο μέσω των τριών παραπάνω πηγών όσο και από το κόστος του προγραμματισμού και των πτήσεων.

1.4.1 Κανόνες Εργασίας και Δομή Αμοιβών

Η πιο στοιχειώδης απόφαση στον προγραμματισμό του πληρώματος είναι η απόφαση για το ποιο πλήρωμα θα εκχωρηθεί σε μια δεδομένη πτήση. Το κόστος μιας τέτοιας εκχώρησης προκύπτει από ένα σύνθετο υπολογισμό. Το πλήρωμα δε λαμβάνει μηνιαίο μισθό, αλλά αντιθέτως αμείβεται για το χρόνο που περνάει στις πτήσεις συν κάποιες επιπρόσθετες αποζημιώσεις για τον επιπλέον χρόνο που περνούν στο έδαφος μεταξύ των πτήσεων και των περιόδων ξεκούρασης. Έτσι, μπορούμε να σκεφτούμε ότι το κόστος μιας πτήσης εξαρτάται από τη διάρκειά της. Επειδή τα πληρώματα για μια δεδομένη οικογένεια αεροσκαφών δεν μπορούν να διακριθούν στο πρόβλημα ταιριάσματος πληρώματος, τα κόστη του πληρώματος συνήθως εκφράζονται σε όρους χρόνου και όχι χρήματος. Ο συνολικός χρόνος πτήσεων είναι σταθερός και προσδιορίζει το χαμηλότερο όριο του άριστου επιπέδου κόστους του πληρώματος. Ο σκοπός του προγραμματισμού του πληρώματος είναι επομένως η ελαχιστοποίηση των πληρωμών και των πιστώσεων, δηλαδή των πληρωμών που γίνονται επιπλέον ή κάτω του κόστους του πραγματικού χρόνου πτήσεων.

1.4.1.1 Περίοδος Υπηρεσιών (Duty Periods)

Μια σειρά κανόνων περιορίζει τους συνδυασμούς πτήσεων που μπορούν να πραγματοποιηθούν από το ίδιο πλήρωμα. Το σύνολο των πτήσεων που μπορούν να πραγματοποιηθούν από ένα πλήρωμα κατά τη διάρκεια μιας ημέρας εργασίας ονομάζεται περίοδος υπηρεσίας (duty period). Αξίζει να σημειωθεί ότι τα μέλη ενός πληρώματος παραμένουν μαζί κατά τη διάρκεια της περιόδου υπηρεσίας τους.

Οι υπηρεσίες συνδέονται άμεσα και με μια σειρά περιορισμών. Ο πιο προφανής από αυτούς είναι ότι οι πτήσεις πρέπει να ακολουθούν η μία την άλλη σε χώρο και χρόνο. Επιπλέον, υπάρχει περιορισμός στον ελάχιστο χρόνο αεργίας μεταξύ δύο συνεχόμενων πτήσεων, γνωστό ως χρόνο σύνδεσης ή χρόνο ξεκούρασης. Υπάρχει όμως και περιορισμός στον μέγιστο χρόνο αεργίας που επιτρέπεται μεταξύ δύο συνεχόμενων πτήσεων. Άλλος ένας περιορισμός είναι ο μέγιστος χρόνος που πρέπει να παρέλθει από περίοδο σε περίοδο υπηρεσίας. Τέλος, αυστηροί κανονισμοί διέπουν τον συνολικό αριθμό των ωρών πτήσεων που το πλήρωμα μπορεί να πραγματοποιήσει κατά τη διάρκεια μιας περιόδου υπηρεσιών.

1.4.1.2 Ζεύγη (Pairings)

Συχνά, η περίοδος μιας υπηρεσίας ξεκινά και σταματά σε διαφορετικά αεροδρόμια. Έτσι, τα μέλη του πληρώματος δεν μπορούν να επιστρέψουν σπίτι τους μετά το τέλος της υπηρεσίας τους και συχνά χρειάζεται να κάνουν στάση κάπου μέχρι να ξεκινήσει η υπηρεσία της επόμενης περιόδου. Τυπικά, το πλήρωμα μπορεί να χρειαστεί να περάσει οπουδήποτε από μία έως και πέντε μέρες στη σειρά μακριά από το σπίτι τους. Μια σειρά από υπηρεσίες και στάσεις ονομάζεται ζεύγος. Γενικά, το πλήρωμα θα παραμείνει μαζί για όλες τις υπηρεσίες μέσα σε ένα ζεύγος.

Υπάρχει μια σειρά λογικών περιορισμών για το τι θα αποτελεί ένα εφικτό συνδυασμό ζεύγους. Σαφώς, η πρώτη υπηρεσία του ζεύγους πρέπει να ξεκινά από τον τόπο διαμονής του πληρώματος, το οποίο ονομάζεται και βάση του πληρώματος (crew base), και η τελευταία υπηρεσία του ζεύγους πρέπει επίσης να τερματίζει εκεί.

Επιπρόσθετα, κάθε περίοδος υπηρεσίας πρέπει να ξεκινά από το ίδιο αεροδρόμιο στο οποίο έληξε η προηγούμενη.

Τα ζεύγη περιορίζονται επιπλέον από μια περίπλοκη διάταξη μεταξύ της απαραίτητης ξεκούρασης, των περιορισμών στις ώρες πτήσης και άλλων περιορισμών. Σε αυτούς τους περιορισμούς περιλαμβάνεται ο μέγιστος αριθμός υπηρεσιών μέσα σε ένα ζεύγος, ο μέγιστος και ο ελάχιστος χρόνος ξεκούρασης μεταξύ των υπηρεσιών και ο μέγιστος χρόνος που παρήλθε από ένα ζεύγος γνωστός και ως χρόνος μακριά από τη βάση (time away from base, TAFB). Στις Η.Π.Α. ένας ιδιαίτερα περίπλοκος περιορισμός είναι ο 8-σε-24 κανόνας που υποδεικνύεται από την ΟΔΑ (Ομοσπονδιακή Διοίκηση Αεροπορίας). Αυτός ο κανόνας απαιτεί επιπλέον ξεκούραση στην περίπτωση που ένα ζεύγος περιλαμβάνει πτήση μεγαλύτερη από 8 ώρες σε οποιαδήποτε 24ωρη περίοδο. Γενικά, είναι επιτρεπτό να υπάρχει πτήση μεγαλύτερη των 8 ωρών σε μια 24ωρη περίοδο μόνο αν ο χρόνος ξεκούρασης που περιλαμβάνεται, δηλαδή ο χρόνος αργίας μεταξύ από δύο περιόδους υπηρεσίας και ο χρόνος ξεκούρασης που ακολουθεί την δεύτερη περίοδο υπηρεσίας, έχει ικανοποιητική διάρκεια.

1.4.1.3 Πρόγραμμα (Schedule)

Όπως ακριβώς μια περίοδος υπηρεσιών είναι μια σειρά από πτήσεις με περιόδους ξεκούρασης στο ενδιάμεσο και ένα ζεύγος είναι μια σειρά από υπηρεσίες με χρόνο αργίας στο ενδιάμεσο, ένα πρόγραμμα είναι απλά μια σειρά από ζεύγη με ρεπό στο ενδιάμεσο. Ωστόσο, μια σημαντική διαφορά-κλειδί μεταξύ των προγραμμάτων και των άλλων δομικών στοιχείων είναι ότι τα προγράμματα σχετίζονται με ατομικά μέλη του πληρώματος και όχι με το συνολικό πλήρωμα. Ο λόγος που συμβαίνει κάτι τέτοιο είναι ότι το κάθε μέλος του πληρώματος έχει διαφορετικές ανάγκες για ρεπό μέσα στην περίοδο του προγράμματος που τυπικά είναι ένας μήνας. Το ρεπό περιλαμβάνει χρόνο για διακοπές, εκπαίδευση κ.ά. Έτσι, κατά τον προσδιορισμό του προγράμματος του πληρώματος πρέπει να ληφθούν υπόψη οι ανάγκες και οι προσωπικές προτιμήσεις του κάθε μέλους του πληρώματος.

Εκτός από τις ανάγκες του κάθε μέλους του πληρώματος, υπάρχουν και περιορισμοί παρόμοιοι με αυτούς που υπάρχουν στις περιόδους υπηρεσίας και στα ζεύγη, για παράδειγμα ο περιορισμός στο μέγιστο μηνιαίο χρόνο πτήσης, στο μέγιστο χρόνο υπηρεσίας το μήνα, στις ελάχιστες συνεχόμενες ημέρες ξεκούρασης, στις ελάχιστες συνολικές ημέρες ξεκούρασης το μήνα, στον ελάχιστο χρόνο ξεκούρασης μεταξύ των υπηρεσιών κ.ό.κ.

Λόγω αυτής της διαφοράς, δεν αποτελεί έκπληξη το γεγονός ότι το κόστος ενός προγράμματος είναι αρκετά διαφορετικό από των άλλων παραγόντων. Ενώ στις υπηρεσίες και στα ζεύγη επικεντρωνόμαστε σε πραγματικά κόστη εργασίας, το κόστος ενός προγράμματος είναι κυρίως συνάρτηση της ικανοποίησης του πληρώματος και του ισορροπημένου φόρτου εργασίας.

1.4.2 Το Πρόβλημα των Ζευγών και της Εκχώρησης Πληρώματος

Το πρόβλημα του προγραμματισμού του πληρώματος χωρίζεται σε δύο υπό-προβλήματα. Πρώτα, πρέπει να λυθεί το πρόβλημα των ζευγών (pairing problem). Σε αυτό το πρόβλημα, πρέπει να γίνει η επιλογή ενός συνόλου ζευγών έτσι ώστε η κάθε πτήση να περιλαμβάνεται σε ένα μόνο ζεύγος και οι πληρωμές και οι πιστώσεις να ελαχιστοποιούνται. Έπειτα, πρέπει να λυθεί το πρόβλημα της εκχώρησης πληρώματος. Σε αυτό το πρόβλημα, τα επιλεγμένα ζεύγη συνδυάζονται με περιόδους ξεκούρασης, διακοπών, εκπαίδευσης και με άλλα διαλείμματα έτσι ώστε να δημιουργηθούν εκτεταμένα ατομικά προγράμματα εργασίας τα οποία, συνήθως, είναι μηνιαία.

1.4.2.1 Το Πρόβλημα των Ζευγών

Το πρόβλημα της δημιουργίας ζευγών σε πτήσεις εσωτερικού λύνεται, συνήθως, σε τρία στάδια: στο ημερήσιο, στις εβδομαδιαίες εξαιρέσεις και στο μεταβατικό.

Το πρώτο στάδιο, το ημερήσιο πρόβλημα, λαμβάνει υπόψιν το σύνολο των πτήσεων οι οποίες πραγματοποιούνται τουλάχιστον τέσσερις φορές την εβδομάδα. Σε αυτό το πρώτο στάδιο, οι πτήσεις αυτές θεωρούμε ότι πραγματοποιούνται κάθε μέρα. Έτσι, πρέπει να βρεθεί το ελάχιστο κόστος των εφικτών ζευγών έτσι ώστε η κάθε πτήση να καλύπτεται ακριβώς μία φορά. Τα ζεύγη σε αυτή τη λύση θεωρούμε ότι επαναλαμβάνονται σε καθημερινή βάση. Το ημερήσιο πρόβλημα αποτελεί μια καλή προσέγγιση μιας και οι περισσότερες πτήσεις εσωτερικού, με ελάχιστες εξαιρέσεις, πραγματοποιούνται κάθε μέρα.

Για ζεύγη που καλύπτουν πολλές ημέρες, θεωρούμε ότι το κάθε πλήρωμα θα εκχωρηθεί σε κάθε μία από τις διαφορετικές υπηρεσίες που υπάρχουν μέσα στο ζεύγος οποιαδήποτε ημέρας. Σε αυτό το σημείο, αξίζει να σημειωθεί ότι η λύση στο εβδομαδιαίο πρόβλημα δεν μπορεί να είναι απόλυτα εφικτή στην πράξη καθώς υποθέτει ότι όλες οι πτήσεις πραγματοποιούνται κάθε μέρα της εβδομάδας. Τα ζεύγη, επομένως, που δεν μπορούν να πραγματοποιηθούν σε συγκεκριμένες ημέρες της εβδομάδας επειδή μία ή περισσότερες πτήσεις δεν είναι προγραμματισμένες για αυτές τις μέρες, αναφέρονται ως σπασμένα ζεύγη (broken pairings).

Το δεύτερο στάδιο για τη δημιουργία ζευγών στο πλήρωμα, το πρόβλημα των εβδομαδιαίων εξαιρέσεων, κατασκευάζει νέα ζεύγη για να διορθώσει τα παραπάνω σπασμένα ζεύγη και για να καλύψει, επίσης, εκείνες τις πτήσεις που πραγματοποιούνται τρεις ή λιγότερες μέρες την εβδομάδα. Έτσι, στο πρόβλημα των εβδομαδιαίων εξαιρέσεων, πρέπει να συνδεθούν οι πτήσεις με συγκεκριμένη ημέρα της εβδομάδας. Επομένως, τα ζεύγη γίνονται πιο συγκεκριμένα με τις ημέρες της εβδομάδας. Συνήθως, χρειάζονται και μεταβατικές πτήσεις, αυτές δηλαδή που θα μεταφέρουν το πλήρωμα ως επιβάτες σε κάποιο άλλο αεροδρόμιο προκειμένου να αναλάβει από εκεί υπηρεσία, προκειμένου να βρεθούν καλές λύσεις για το πρόβλημα των εβδομαδιαίων εξαιρέσεων. Συνδυάζοντας το ημερήσιο πρόγραμμα και τις εβδομαδιαίες εξαιρέσεις, οι λύσεις στο πρόβλημα της δημιουργίας ζευγών καλύπτουν κάθε πτήση του εβδομαδιαίου προγράμματος ακριβώς μία φορά.

Τέλος, πρέπει να σημειωθεί ότι οι αεροπορικές εταιρείες διαφοροποιούν τις πτήσεις που πραγματοποιούν σε τακτική βάση, συχνά κάθε τρίμηνο, και, σε κάποιο βαθμό, ακόμα και μηνιαία. Έτσι, τα ζεύγη που περιλαμβάνουν πολλές ημέρες

μπορούν να αποδειχθούν προβληματικά στο τέλος του μηνιαίου προγράμματος πτήσεων. Για παράδειγμα, την τελευταία μέρα του μήνα ένα νέο πλήρωμα πρέπει να ξεκινήσει το κάθε ζεύγος για να καλύψει τις πτήσεις αυτής της ημέρας. Παρόλα αυτά, οι υπολειπόμενες ημέρες του ζεύγους μπορεί να μην είναι διαθέσιμες μιας και οι πτήσεις που μπορεί να πραγματοποιηθούν σύμφωνα με το πρόγραμμα του επόμενου μήνα να είναι διαφορετικές. Πρέπει, επομένως, να λυθεί ένα τρίτο στάδιο του προβλήματος ζευγών, το μεταβατικό πρόβλημα. Αυτό το πρόβλημα, κατασκευάζει έτσι τα ζεύγη ώστε να καλύπτουν πτήσεις για μικρό περιθώριο ημερών για να μην επηρεάζεται από την αλλαγή των πτήσεων από μήνα σε μήνα.

Σε όλα αυτά τα προβλήματα, γίνεται έμφαση στο ότι ο κύριος σκοπός είναι η ελαχιστοποίηση των πληρωμών και των πιστώσεων (pay-and-credit), δηλαδή του κόστους της εργασίας που είναι πάνω αλλά και κάτω του ελάχιστου χρόνου πτήσεων που απαιτείται.

1.4.2.2 Το Πρόβλημα Εκχώρησης Πληρώματος

Δεδομένης της λύσης στο πρόβλημα δημιουργίας ζευγών, δηλαδή του συνόλου των ζευγών που προσφέρουν το ελάχιστο κόστος και καλύπτουν όλες τις πτήσεις σε μηνιαία βάση, πρέπει να γίνει η ανάθεση των πτήσεων σε συγκεκριμένα άτομα αυτών των ζευγών. Αυτό απαντάται στο πρόβλημα ανάθεσης πληρώματος.

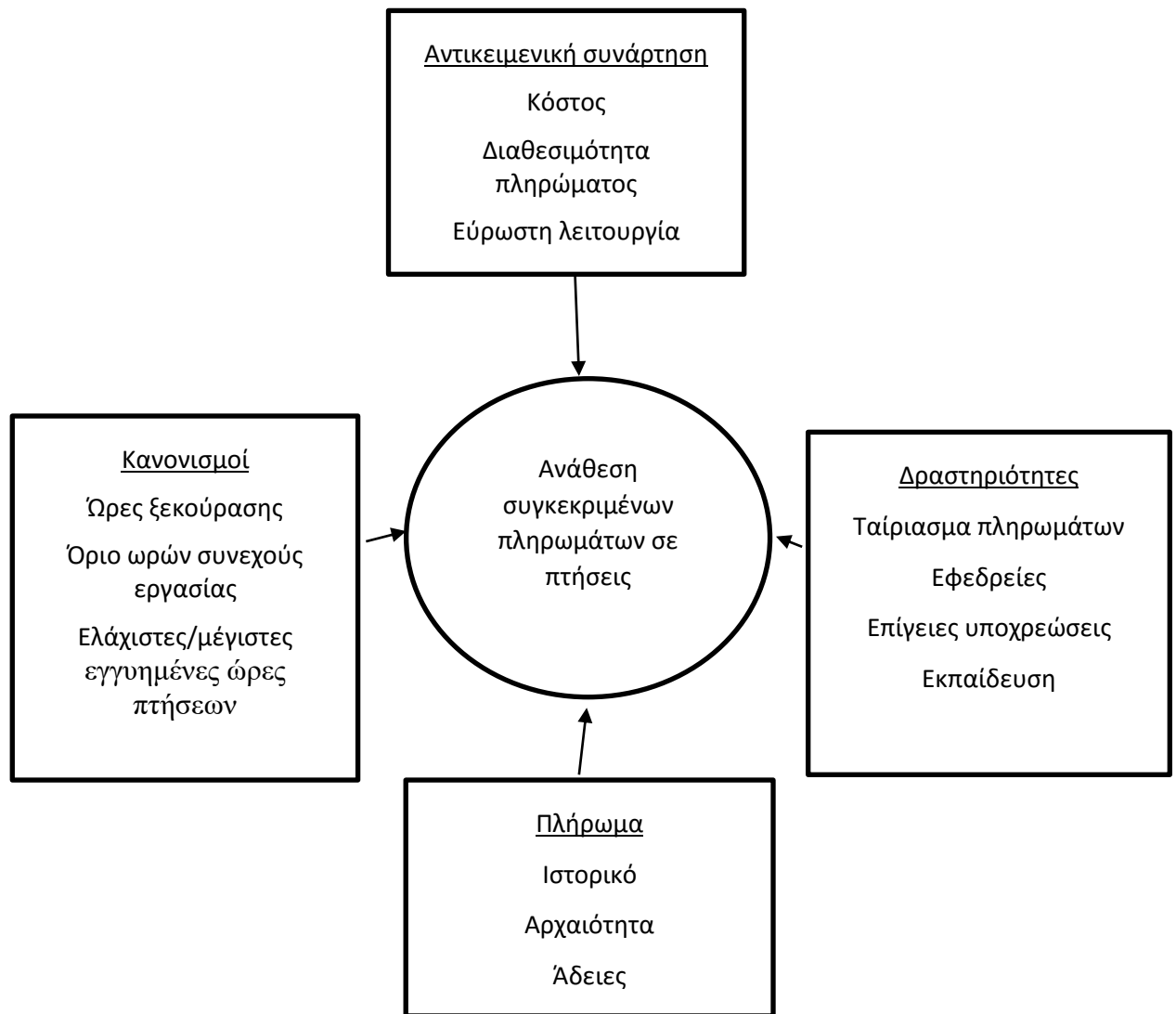
Έτσι όπως το πρόβλημα των ζευγών επιλέγει ένα ελάχιστο κόστος συνόλου ζευγών έτσι ώστε κάθε πτήση να καλύπτεται, το πρόβλημα ανάθεσης πληρώματος επιλέγει ένα σύνολο προγραμμάτων (μια αλυσίδα συνεχόμενων ζευγών που ικανοποιούν ένα πλήθος κανονισμών) έτσι ώστε το κάθε ζεύγος να καλύπτεται. Στο πλαίσιο αυτό, το κάθε ζεύγος αντιστοιχεί σε συγκεκριμένες ημέρες του προγράμματος.

Παρά τις ομοιότητες, αυτά τα δύο προβλήματα απαντώνται ξεχωριστά, τόσο στη βιομηχανία όσο και στην ακαδημαϊκή βιβλιογραφία. Δύο είναι οι κύριοι λόγοι που συμβαίνει κάτι τέτοιο. Πρώτον, στο πρόβλημα ζευγών του πληρώματος εκχωρούμε ολόκληρο το πλήρωμα σε μια πτήση, ενώ στο πρόβλημα εκχώρησης προσωπικού το

πλήρωμα προγραμματίζεται ατομικά με κάθε ένα ζεύγος να καλύπτεται από πολλαπλά μέλη πληρώματος. Δεύτερον, το πρόβλημα των ζευγών επικεντρώνεται στην ελαχιστοποίηση του κόστους εργασίας ενώ το πρόβλημα εκχώρησης προσωπικού δίνει μεγαλύτερη έμφαση στην ικανοποίηση των αναγκών του πληρώματος και στην εύρεση ενός ισορροπημένου προγράμματος εργασίας.

Το πρόβλημα ανάθεσης προσωπικού λύνεται σε δύο στάδια. Στο πρώτο στάδιο, κατασκευάζεται ένα σύνολο προγραμμάτων έτσι ώστε το κάθε ζεύγος να περιλαμβάνεται σε ακριβώς τόσα προγράμματα όσο χρειάζονται για να είναι πλήρως στελεχωμένη η πτήση. Έπειτα, στο δεύτερο στάδιο, αυτά τα προγράμματα αναθέτονται σε ατομικά μέλη του πληρώματος. Η ανάθεση των ζευγών πληρωμάτων με πτήσεις γίνεται σε συνδυασμό με άλλες δραστηριότητες όπως τα επίγεια καθήκοντα, τα καθήκοντα εφεδρείας σε περίπτωση ανάγκης, καθώς και οι εκτός εργασίας φραγμοί όπως άδειες ή οποιαδήποτε άλλα προσωπικά προβλήματα που περιορίζουν τη δυνατότητα του εργαζομένου να παρευρεθεί στη δουλειά του συγκεκριμένες ώρες. Τα ζεύγη τοποθετούνται διαδοχικά σε λίστες και ανατίθενται σε μεμονωμένα μέλη του πληρώματος ονομαστικά πλέον. Η ανάθεση αυτή γίνεται συνήθως για κάθε μήνα ξεχωριστά και κάθε λίστα περιέχει, συνήθως, κάποια ιστορικά στοιχεία (π.χ. ημερομηνίες αδειών) που βοηθούν στον προγραμματισμό επόμενων μηνών. Και σε αυτό το πρόβλημα, σύνθετοι κανόνες και οι κανονισμοί που προέρχονται από τη νομοθεσία και τις συμβατικές συμφωνίες πρέπει να καλυφθούν από τις λύσεις στην προσπάθεια βελτίωσης κάποιας αντικειμενικής συνάρτησης που αφορά τη λειτουργία και το κέρδος της εταιρείας.

Για την επίλυση του προβλήματος ανάθεσης πληρωμάτων έχει προτιμηθεί να γίνεται ξεχωριστά η ανάθεση για κάθε ειδικότητα, για παράδειγμα συνήθως οι πιλότοι έχουν συγκεκριμένη εκπαίδευση σε ένα ειδικό τύπο αεροσκάφους και έτσι θα μπορούν να πετάνε μόνο με αυτόν τον τύπο, σε αντίθεση με τους αεροσυνοδούς που έχουν την δυνατότητα να εργασθούν σε διαφορετικού τύπου αεροσκάφη. Χρησιμοποιώντας ακέραιο προγραμματισμό μπορεί να γίνει η μεγιστοποίηση της συμβολής κάθε μέλους του πληρώματος στο πρόγραμμα πτήσεων αλλά και η ελαχιστοποίηση του κόστους της αεροπορικής εταιρείας.



Σχήμα 1.2 : Διάγραμμα του προβλήματος ανάθεσης πληρώματος σε πτήσεις

ΚΕΦΑΛΑΙΟ 2

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

2.1 Εισαγωγή

Ο προγραμματισμός του πληρώματος των αεροπορικών εταιριών έχει αποτελέσει αντικείμενο συστηματικής μελέτης τις τελευταίες δεκαετίες και καλύπτει ένα μεγάλο μέρος της βιβλιογραφίας που ασχολείται με τα προβλήματα που καλούνται να αντιμετωπίσουν οι αεροπορικές εταιρίες. Το πρόβλημα επιλύεται με τη βοήθεια ενός μαθηματικού μοντέλου και με τη χρησιμοποίηση του γραμμικού και ακέραιου προγραμματισμού καθώς και διάφορων τεχνικών όπως η μέθοδος branch-and-price. Βασικός στόχος είναι η ελαχιστοποίηση του κόστους, με παράλληλη ικανοποίηση των εκάστοτε περιορισμών.

2.2 Ανάθεση Στόλου (Fleet Assignment)

Το πρόβλημα της ανάθεσης στόλου καθορίζει τον τύπο του αεροσκάφους, ο οποίος θα ανατεθεί σε κάθε πτήση, δεδομένου του προγράμματος των πτήσεων, ενώ παράλληλα πρέπει να ικανοποιεί ορισμένους περιορισμούς. Το πρόβλημα της ανάθεσης του στόλου μοντελοποιήθηκε για πρώτη φορά από τους Hane et al. (1995), οι οποίοι επίλυσαν ένα πρόβλημα έχοντας 11 στόλους και 2500 πτήσεις. Το πρόβλημα απλοποιήθηκε κάνοντας την υπόθεση ότι κάθε πτήση πραγματοποιείται κάθε ημέρα της εβδομάδας. Σε συνέχεια της δουλειά τους, οι Clark et al. (1996) γενίκευσαν το βασικό μοντέλο ανάθεσης στόλου, συμπεριλαμβάνοντας τη συντήρηση του στόλου των αεροσκαφών και τον προγραμματισμό του πληρώματος.

Οι Berge και Hopperstand (1993) και οι Hane et al. (1995) χρησιμοποίησαν τη δομή ενός δικτύου χωροχρόνου, το οποίο αποτελεί και το κύριο πλαίσιο για την μορφοποίηση του προβλήματος της ανάθεσης του στόλου στη σημερινή εποχή. Παρόλα αυτά, όπως υπέδειξαν οι Rushmeir and Kontogiorgis (1997), το δίκτυο χωροχρόνου δεν είναι ικανό να διακρίνει τα διάφορα είδη αεροσκαφών στο έδαφος, κάτι το οποίο περιορίζει την εφαρμογή του στο πρόβλημα της συνεχόμενης δρομολόγησης αεροσκαφών.

Οι Rushmeir and Kontogiorgis (1997) μοντελοποίησαν το πρόβλημα ανάθεσης στόλου ως ένα μικτό ακέραιο πρόβλημα ροής, όπου το προϊόν είναι ο στόλος και οι περιορισμοί του προβλήματος ικανοποιούν τις λειτουργικές απαιτήσεις και την κάλυψη όλων των πτήσεων.

Οι Lohatepanont and Barnhart (2004) ασχολήθηκαν με το πρόβλημα του μεγάλου μεγέθους και της μεγάλης πολυπλοκότητας, κάνοντας βαθμιαίες προσεγγίσεις στον σχεδιασμό του προγράμματος και δουλεύοντας σε κάθε βήμα με ένα υποσύνολο των πτήσεων από το συνολικό πρόγραμμα. Οι Kim and Barnhart (2005) ασχολήθηκαν με το πρόβλημα σχεδιασμού του προγράμματος των πτήσεων για αεροπορική charter με την ανάπτυξη ευρετικών ακολουθιών λύσεων και συγκρίνοντας τα αποτελέσματά που προέκυπταν κάθε φορά.

Επιπλέον, οι Sherali and Zhu (2007) εισήγαγαν μία στοχαστική προσέγγιση δύο σταδίων, όπου η αρχική ανάθεση στόλου γίνεται στο πρώτο στάδιο και στο δεύτερο στάδιο βελτιώνεται η πρόβλεψη της ζήτησης των αεροσκαφών και γίνεται η ανάθεση στόλου σύμφωνα με τον τύπο του κάθε αεροσκάφους.

2.3 Δρομολόγηση αεροσκαφών (Aircraft Routing)

Η λύση που προκύπτει από το πρόβλημα της ανάθεσης του στόλου, υποδεικνύει τη ροή των αεροσκαφών, σύμφωνα με το είδος του αεροσκάφους, μέσα στο δίκτυο πτήσεων. Η δρομολόγηση αεροσκαφών είναι η διαδικασία, στην οποία αναγνωρίζεται από ποιο στόλο ή οικογένεια αεροσκαφών προέρχεται το κάθε

αεροσκάφος, στο οποίο ανατίθεται μια πτήση. Το πρόβλημα της δρομολόγησης αεροσκαφών αναπτύχθηκε από τους Daskin and Panayotopoulos (1989), Kabbani and Patty (1992), Talluri (1996,1998), Clarke et al. (1997), Gopalan and Talluri (1998b) και Barthlomew-Biggs et al. (2003). Οι περισσότερες μέθοδοι επίλυσης είναι ευρετικοί. Για παράδειγμα, οι Kabbani and Patty (1992) μοντελοποίησαν το πρόβλημα δρομολόγησης των αεροσκαφών σαν ένα σύνολο επιμέρους προβλημάτων. Επιπλέον, οι Clarke et al. (1997) παρουσίασαν ένα μοντέλο βασισμένο πάνω στις πτήσεις και περιέγραψαν μια λαγκρατζιανή προσέγγιση της λύσης, η οποία προσθέτει τους περιορισμούς συντήρησης, όταν αυτό είναι αναγκαίο. Σχετικά πιο πρόσφατα, οι Bartholomew-Biggs et al. (2003) μελέτησαν το πρόβλημα της δρομολόγησης των αεροσκαφών, το οποίο είναι σχετικό με την εύρεση της άριστης διαδρομής για τη μείωση της απόστασης μεταξύ της αρχής και του προορισμού, ενώ παράλληλα αποφεύγονται τυχόν εμπόδια. Άμεσες μελέτες, ντετερμινιστικές και στοχαστικές μέθοδοι αριστοποίησης έχουν σχεδιαστεί και τα σχετικά αποτελέσματα έχουν συγκριθεί για να βρεθεί η άριστη δρομολόγηση του στόλου των αεροσκαφών.

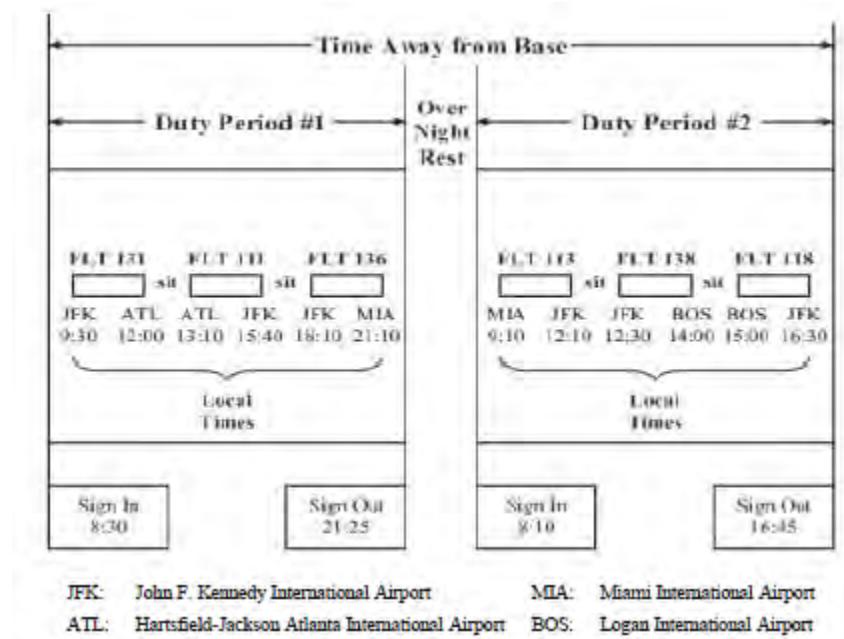
2.4 Προγραμματισμός Πληρώματος (Crew Scheduling)

Ο προγραμματισμός του πληρώματος περιλαμβάνει την αναγνώριση της ακολουθίας των πτήσεων και την ανάθεση του πληρώματος σε αυτές και παρόμοια με τη δρομολόγηση των αεροσκαφών, πραγματοποιείται έπειτα από την ανάθεση του στόλου.

Το πρόβλημα του προγραμματισμού του πληρώματος λύνεται συνήθως σε δύο συνεχόμενες φάσεις: τη φάση του ταιριάσματος του πληρώματος για τη δημιουργία ζευγών, crew pairing, και τη φάση της ανάθεσης, crew rostering. Αυτό γίνεται κυρίως γιατί τα δύο προβλήματα είναι δύσκολο να αντιμετωπιστούν ταυτόχρονα, όπως αναφέρει και ο Massoud Bazargan (2010).

Ο προγραμματισμός του πληρώματος παρουσιάζεται συνήθως ως ένα σύνολο μοντέλων, στα οποία οι στήλες παρουσιάζουν τα εφικτά ζεύγη και οι γραμμές τις προγραμματισμένες πτήσεις. Σύμφωνα με του Minoux (1984) και Lavoie et al. (1988), επειδή ένα τέτοιο μοντέλο περιλαμβάνει ένα μεγάλο αριθμό μεταβλητών, η κύρια προτεινόμενη προσέγγιση της λύσης είναι βασισμένη στην τεχνική του column generation, η οποία, σύμφωνα με τους Ribeiro et al. (1989), Gamache et al. (1994), Desaulniers et al. (1997a), Vance et al. (1997a), Barnhart and Shenoι (1998), Barnhart et al. (1998b), Gamache et al. (1998), Stojkovic et al. (1998), Lettovsky et al. (2000), and Yan and Chang (2002), συνδυάζεται με τη διαδικασία του branch-and-price. Η μέθοδος branch-and-cut προτάθηκε επίσης από τους Hoffman and Padberg (1993) και μια λαγκρασιανή χαλάρωση ευρετικά βασισμένη υιοθετήθηκε από τους Alefragis et al. (2000).

Όπως έχει αναφερθεί, ένα ζεύγος αποτελείται από μια ακολουθία πτήσεων ή πιο συγκεκριμένα είναι μια ακολουθία υπηρεσιών, που συνδέουν δύο ή περισσότερες ημέρες εργασίας και χωρίζονται από περιόδους ξεκούρασης. Ένα ζεύγος πρέπει να ξεκινά και να τελειώνει στην ίδια βάση πληρώματος σύμφωνα με τους Balaji, Gopalakrishnan and Johnson (2005) και Massoud (2010). Για παράδειγμα, στην εικόνα 2.1 (Massoud (2010)) παρουσιάζεται ένα ζεύγος δύο ημερών και δείχνει τις περιόδους υπηρεσιών, τις συνδέσεις μεταξύ των ζευγών, τις ολονύχτιες παύσεις και τις ώρες έναρξης και τερματισμού, υποθέτοντας ως βάση του πληρώματος το αεροδρόμιο JFK (αεροδρόμιο John F.Kennedy). Όπως φαίνεται στο διάγραμμα, το πλήρωμα μένει για μια νύχτα, μακριά από τη βάση του, και σε αυτή την περίπτωση, όπως θα φανεί και στις δομές του κόστους, η αεροπορική εταιρία πρέπει να πληρώσει για το δείπνο, την διαμονή, τις μεταφορές κ.ά. του πληρώματος.



Εικόνα 2.1: Ένα τυπικό ζεύγος μέσα σε μια περίοδο υπηρεσίας

Ο αντικειμενικός σκοπός του προβλήματος ταιριάσματος του πληρώματος είναι η εύρεση ενός συνόλου ζευγών, έτσι ώστε σε κάθε πτήση να περιλαμβάνεται ακριβώς ένα ζεύγος και το συνολικό κόστος του πληρώματος να ελαχιστοποιείται σύμφωνα με τους Balaji, Gopalakrishnan and Johnson (2005) και Massoud (2010). Προκειμένου να επιτευχθεί κάτι τέτοιο, τα εφικτά (ή τα νόμιμα) ζεύγη πρέπει να δημιουργηθούν πρώτα και στην συνέχεια τα εξαχθέντα ζεύγη υπόκεινται σε αριστοποίηση, προκειμένου να λυθεί το πρόβλημα. Αυτά τα δύο στάδια είναι εξίσου σημαντικά και προφανώς κανένα από τα δύο δεν μπορεί να παραληφθεί.

Στο στάδιο δημιουργίας των ζευγών, τα ζεύγη δημιουργούνται σύμφωνα με συγκεκριμένους κανόνες και περιορισμούς, όπως για παράδειγμα ότι πρέπει να ξεκινούν και να τερματίζουν στην ίδια βάση του πληρώματος και να λαμβάνουν υπόψη τιμές, όπως τον αριθμό των μέγιστων υπηρεσιών, τον ελάχιστο χρόνο ξεκούρασης, τον μέγιστο χρόνο πτήσης κλπ. Μιας και όλα τα εφικτά ζεύγη παράγονται σε αυτό το στάδιο, ο αριθμός των ζευγών είναι συνήθως πολύ μεγάλος σύμφωνα με τους Balaji, Gopalakrishnan and Johnson (2005), ειδικά για μεγάλες αεροπορικές εταιρίες. Έτσι, η διαδικασία εύρεσης των ζευγών πρέπει να

περιλαμβάνει και επιπλέον κανονισμούς και φίλτρα, ώστε να επιλέγονται μόνο τα υποθετικά καλά ζεύγη.

Όσον αφορά το στάδιο της αριστοποίησης, ένας αριθμός ζευγών επιλέγεται από το σύνολο όλων των πιθανών ζευγών, έτσι ώστε τα όλα τα επιλεγόμενα ζεύγη να καλύπτουν όλες τις διαθέσιμες πτήσεις και ταυτόχρονα να επιδιώκουν ελαχιστοποίηση του κόστους, με το οποίο συνδέονται.

Πρέπει, επίσης, να σημειωθεί ότι κατά τη διαδικασία δημιουργίας ζευγών δεν έχει δημιουργηθεί κανένα ατομικό πρόγραμμα του πληρώματος ακόμα, παρά μόνο γενικά άριστα ζεύγη, τα οποία μπορούν να συνδυαστούν μεταξύ τους μαζί με άλλα καθήκοντα για την ανάθεση σε συγκεκριμένα μέλη του πληρώματος στην επόμενη φάση.

2.4.1 Παράλληλες Προσεγγίσεις στο Πρόβλημα Ταιριάσματος Πληρώματος (Crew Pairing)

Το πρόβλημα ταιριάσματος του πληρώματος μπορεί να χρειαστεί 10 έως 20 ώρες CPU για να επιλυθεί έτσι προσεγγιστικά. Αυτό μπορεί να δημιουργήσει πολλά προβλήματα αν προστεθούν και οι υποθετικές αναλύσεις “what-if”. Για να μειωθεί ο χρόνος υπολογισμού εφαρμόζεται η χρησιμοποίηση παράλληλων αλγορίθμων για τη δημιουργία ζευγών. Παρόλα αυτά, το πρόβλημα δημιουργίας ζευγών είναι ένα πρόβλημα ακέραιου προγραμματισμού και μαζί με τους παράλληλους αλγόριθμους είναι δύσκολο να δουλέψει. Έτσι, ο σχεδιασμός των παράλληλων αλγορίθμων για το πρόβλημα ταιριάσματος του πληρώματος είναι ένα απαιτητικό πρόβλημα, το οποίο άρχισε να ερευνάται σε μεγαλύτερο βαθμό τα τελευταία χρόνια.

Η βασική ιδέα των παράλληλων αλγορίθμων για τη δημιουργία ζευγών είναι η διανομή των πτήσεων, οι οποίες ξεκινούν από τη βάση του πληρώματος, μεταξύ των επεξεργαστών, με τον κάθε επεξεργαστή να απαριθμεί όλα τα ζεύγη που ξεκινούν με την πτήση που έχει ανατεθεί. Οι Goumopoulos et al. (1997) πρότειναν ένα αλγόριθμο βασισμένο στο παράδειγμα αφεντικού/εργάτη όπου το αφεντικό διανείμει της πτήσεις μία προς μία στους εργάτες. Όταν ο εργάτης μείνει άεργος τότε ζητά από

το αφεντικό μία νέα πτήση να ξεκινήσει. Οι Klabjan and Schwan (2000) κατήργησαν τα «αφεντικά» με τους επεξεργαστές να εναλλάσσουν τις εργασίες μεταξύ τους.

Σε άλλες μελέτες, οι Alefragis et al. (1998), Sanders et al. (1999) και Alefragis et al. (2000) επικεντρώθηκαν στον παραλληλισμό της απαρίθμησης των ζευγών με τον λαγκρασιανό αλγόριθμο του Andersson et al. (1998). Μια εντελώς διαφορετική προσέγγιση δόθηκε από τους Klabjan et al. (2001b), όπου η γραμμική χαλάρωση επιλύεται παράλληλα με τη δημιουργία παράλληλων ζευγών.

Ο πιο υποσχόμενος αλγόριθμος για τον παραλληλισμό είναι ο branch-and-bound μιας και η χαμηλή διακριτότητα είναι εύκολα επιτεύξιμη, αποκλείοντας του κόμβους του branch-and-bound. Οι Gedron and Crainic (1994) έκαναν μια έρευνα για τους παράλληλους αλγόριθμους branch-and-bound. Οι Klabjan et al. (2001) περιγράφουν ένα παράλληλο branch-and-price αλγόριθμο.

2.4.2 Παράλληλες Προσεγγίσεις στην Ανάθεση Πληρώματος (Crew Rostering)

Μια προσέγγιση για το πρόβλημα της ανάθεσης πληρώματος περιγράφεται από τους Campbell, Durfee, and Hines (1997), Christou et al. (1999), Jarrah and Diamond (1997). Οι δύο πρώτες από αυτές τις αναφορές περιγράφουν εφαρμογές στις εταιρίες FedEx και Delta Air Lines. Το πρόβλημα λύνεται χρησιμοποιώντας μετά-ευρετικούς αλγορίθμους. Οι Jarrah and Diamond (1997) ανέπτυξαν ένα ημιαυτόματο σύστημα, το οποίο βασίζεται στο column generation. Το κέντρο ερευνών GERAD ανέπτυξε μία προσέγγιση με την τεχνική column generation, όπου τα υπο-προβλήματα που κάνουν την ανάθεση στα άτομα, επιλύονται ως οι συντομότεροι δρόμοι υπό περιορισμό.

Η ανάθεση του πληρώματος καμπίνας της Air New Zealand αναλύθηκε από τους Day and Ryan (1997). Το πρόβλημα αναλύεται ως πρόβλημα ανάθεσης των ημερών ξεκούρασης ακολουθούμενο από το πρόβλημα της ανάθεσης ζευγών και άλλων δραστηριοτήτων. Και τα δύο προβλήματα λύνονται με column generation.

Μια προσέγγιση ακέραιου προγραμματισμού παρουσιάστηκε από τους Capranera and Gallo (2001) και εφαρμόστηκε για την επίλυση μικρών περιπτώσεων

για μεσαίου μεγέθους εταιρίες. Οι El Moudani et al. (2001) προτείνουν μία ευρετική προσέγγιση για την επίλυση του προβλήματος σε αεροπορικές εταιρίες μεγαλύτερου μεγέθους.

Τα άρθρα των Dawid, König, and Strauss (2001), König and Strauss (2000a),(2000b) εξετάζουν μια διαφορετική πτυχή μοντελοποίησης για την ανάθεση πληρώματος και περιγράφουν μια εκτενή ευρετική τεχνική διάδοσης.

2.5 Μαθηματικές τεχνικές προγραμματισμού

Σύμφωνα με τη μαθηματική του μορφή, το πρόβλημα ταιριάσματος πληρώματος πρέπει να θεωρηθεί ως ένα μοντέλο Δυναδικού Ακέραιου Προγραμματισμού (Binary Integer Programming, BIP). Σε προβλήματα της πραγματικής ζωής, αυτά τα μοντέλα περιλαμβάνουν συνήθως μεγάλο αριθμό μεταβλητών. Τρεις είναι οι πιο γνωστές προσεγγίσεις για μεγάλα προβλήματα, η branch-and-bound, branch-and-cut & branch-and-price και η column generation.

2.5.1 Branch-and-Bound

Για όλα τα προβλήματα ακέραιου προγραμματισμού, η προσέγγιση branch-and-bound ξεκινά με την επίλυση του γραμμικού προβλήματος χαλάρωσης. Έπειτα, χρησιμοποιώντας τη λύση την οποία αποκομίζει, η οποία είναι πολύ πιθανό να μην είναι ακέραια, δημιουργούνται δύο υπο-προβλήματα προκειμένου να περιοριστεί το εύρος των λύσεων. Οι λύσεις από αυτά τα υπο-προβλήματα οριοθετούν τη λύση του αρχικού προβλήματος. Χρησιμοποιώντας συγκεκριμένους κανόνες η διαδικασία συνεχίζεται έως ότου επιτευχθεί μία άριστη ακέραια λύση. Σύμφωνα με τον Potter (2008) , οι διακλαδώσεις μπορούν να είναι είτε μεταβλητές, είτε περιορισμοί, οι οποίοι μπορούν να επηρεάσουν τόσο τη δυσκολία επίλυσης των υπο-προβλημάτων

όσο και το πόσο καλά μπορεί η προσέγγιση να λειτουργεί. Επομένως, γίνεται εύκολα αντιληπτό ότι ο καθορισμός της σειράς των διακλαδώσεων είναι κρίσιμος.

2.5.2 Branch and Price

Η σωστή επίλυση μεγάλων προβλημάτων ακέραιου προγραμματισμού απαιτεί την μοντελοποίησή τους με τέτοιον τρόπο, ώστε το πρόβλημα γραμμική χαλάρωσή να δίνει λύσεις με πολύ καλή προσέγγιση στο σύνολο των εφικτών λύσεων του αρχικού προβλήματος. Γι'αυτό και τις τελευταίες δεκαετίες έχει δοθεί μεγάλη σημασία στην μέθοδο branch and cut. Οι Hoffman και Padberg (1985), καθώς και οι Nemhauser και Wolsey (1988) έδωσαν μεγάλη σημασία στην μέθοδο αυτή και έκαναν μια αρχική γενική παρουσίασή της.

Σύμφωνα με τους Barnhart, Johnson, Nemhauser, Savelsberg και Vance (1996) η κύρια ιδέα αυτής της μεθοδολογίας είναι απλή. Ένα μέρος των περιορισμών μένει έξω από το πρόβλημα γραμμικής χαλάρωσης (linear programming relaxation problem), επειδή υπάρχουν πάρα πολλοί ώστε να τους χειριστεί όλους αποτελεσματικά. Πολλοί από αυτούς τους περιορισμούς πολύ πιθανόν να μην επηρεάσουν και καθόλου την βέλτιστη λύση του προβλήματος. Έτσι, εάν η βέλτιστη λύση του προβλήματος γραμμικής χαλάρωσης είναι μη εφικτή για το αρχικό πρόβλημα, τότε ένα νέο υπο-πρόβλημα λύνεται, προκειμένου να προσδιοριστούν οι ανισότητες-περιορισμοί, που παραβιάζονται. Εάν βρεθούν ένας ή περισσότεροι περιορισμοί, που να παραβιάζονται, ορισμένοι από αυτούς εισάγονται στο πρόβλημα γραμμικής χαλάρωσης, προκειμένου να λυθεί το πρόβλημα της μη εφικτής λύσης. Η μέθοδος branch and cut, η οποία είναι ένας συνδυασμός της μεθόδου διακλάδωσης και οριοθέτησης (branch and bound) με γραμμική χαλάρωση, επιτρέπει την διαδικασία αυτή να γίνεται σε όλους τους κλάδους του δέντρου της μεθόδου branch and bound.

Η μέθοδος branch and price δεν διαφέρει και πολύ από την μέθοδο branch and cut. Η διαφορά τους έγκειται στο γεγονός ότι στο πρόβλημα της γραμμική χαλάρωση αντί να περιορίζουμε τον αριθμό των ανισώσεων, περιορίζουμε τον αριθμό των μεταβλητών.

Στην branch and price μέθοδο, επειδή οι μεταβλητές που μπορεί να προκύψουν είναι πάρα πολλές, ένα μέρος από αυτές λαμβάνεται υπόψη και μετέχουν στο πρόβλημα γραμμικής χαλάρωσης. Αυτό συμβαίνει διότι, λόγω του μεγάλου αριθμού των μεταβλητών δεν μπορεί να γίνει αποτελεσματική διαχείριση όλων. Επίσης, πολλοί από αυτούς, μπορεί να μην επηρεάσουν και καθόλου την λύση, καθώς η τιμή τους μπορεί να είναι μηδέν. Στην συνέχεια για να γίνει ο έλεγχος εάν η λύση του προβλήματος γραμμικής χαλάρωσης είναι βέλτιστη, λύνεται ένα νέο υποπρόβλημα, το οποίο λέγεται column generation subproblem. Το υποπρόβλημα αυτό, χρησιμοποιεί τις δυικές τιμές (duals) της λύσης του προβλήματος της γραμμικής χαλάρωσης, προκειμένου να βρει μεταβλητές που μπορούν να εισαχθούν στο πρόβλημα. Εάν τέτοιες μεταβλητές βρεθούν, εισάγονται στο πρόβλημα γραμμικής χαλάρωσης, το οποίο και λύνεται ξανά, ώστε να δώσει καλύτερη βέλτιστη λύση. Η διακλάδωση στο δέντρο αρχίζει, όταν το column generation subproblem δεν μπορεί να βρει άλλες μεταβλητές για να εισαχθούν στο πρόβλημα και η λύση του προβλήματος γραμμικής χαλάρωσης δεν ικανοποιεί τις συνθήκες του αρχικού προβλήματος.

Εκ πρώτης όψεως μπορεί η μέθοδος αυτή να μη δείχνει κάτι περισσότερο από το συνδυασμό της μεθόδου του column generation και της μεθόδου branch and bound για την επίλυση προβλημάτων γραμμικού προγραμματισμού. Ωστόσο ο Appelgren (1969) παρατήρησε ότι τα πράγματα δεν ήταν και τόσο απλά. Υπάρχουν πολύ μεγάλες δυσκολίες στην επίλυση προβλημάτων ακέραιου προγραμματισμού με την μέθοδο του column generation.

Κατά καιρούς εμφανίστηκαν στην βιβλιογραφία διάφορες μορφές της μεθόδου branch and price, οι οποίες είχαν αναπτυχτεί για την επίλυση συγκεκριμένων προβλημάτων. Οι Desrosiers, Dumas, Solomon και Soumis (1995) αναφέρουν την ευρεία περιοχή εφαρμογής της μεθόδου αυτής σε προβλήματα προγραμματισμού προσωπικού και μέσων μεταφοράς.

2.5.3 Column Generation

Η προσέγγιση του Column Generation επίσης επιδιώκει τη διαχείριση ενός μικρότερου προβλήματος, με ένα μικρότερο υποσύνολο μεταβλητών, παρά τη διαχείριση ενός σημαντικά μεγαλύτερου συνόλου μεταβλητών. Όπως περιγράφεται εκτενώς από τον Shenoί (1994), ένα περιορισμένο κύριο πρόβλημα δημιουργείται από το αρχικό πρόβλημα αριστοποίησης του πληρώματος, χρησιμοποιώντας ένα συγκεκριμένο αριθμό μεταβλητών, που συνιστά μια εφικτή λύση. Αυτό το νέο υπο-πρόβλημα επιλύεται και υπολογίζονται οι εξαγόμενες δυικές τιμές και έπειτα σύμφωνα με συγκεκριμένους υπολογισμούς καθορίζεται αν θα προστεθούν περισσότερες στήλες, προκειμένου να επιτευχθεί η άριστη λύση του προβλήματος. Αυτή η διαδικασία επαναλαμβάνεται έως ότου δεν υπάρχουν άλλες στήλες, οι οποίες να μπορούν να προστεθούν.

Σύμφωνα με τους Florez et al. (2009), αυτή η συνήθης μέθοδος του column generation μπορεί να αναφερθεί και ως δυναμική μέθοδος column generation. Παρόλα αυτά, αν το περιορισμένο κύριο πρόβλημα είναι ένα πρόβλημα γραμμικής χαλάρωσης του αρχικού προβλήματος, οι ακέραιες λύσεις δεν μπορούν να εγγυηθούν στο τέλος.

ΚΕΦΑΛΑΙΟ 3

ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ

3.1 Εισαγωγή

Το πρόβλημά μας, όπως έχουμε αναφέρει και παραπάνω είναι πρόβλημα crew scheduling. Για την επίλυσή του είναι αναγκαίο να γίνει η ανάλυσή του αρχικά, ώστε να διασαφηνιστούν οι διάφοροι περιορισμοί που το διέπουν, καθώς και ο αντικειμενικός του σκοπός, δηλαδή το τι αποσκοπεί το πρόβλημα να βελτιστοποιήσει. Αφού ολοκληρωθεί η ανάλυσή του, στην συνέχεια θα πρέπει να μοντελοποιηθεί μαθηματικά, δηλαδή να γραφτούν οι εξισώσεις, οι οποίες αποτελούν και την μαθηματική του περιγραφή. Η μέθοδος, η οποία θα ακολουθηθεί για την επίλυσή του είναι η μέθοδος branch and price. Η μέθοδος αυτή, όπως έχουμε αναφέρει, αποτελείται από το master problem και το column generation sub-problem. Παρακάτω για καθένα από αυτά τα δύο προβλήματα αρχικά θα γίνεται μια γενική περιγραφή των εξισώσεων, οι οποίες τα περιγράφουν και στην συνέχεια θα γίνεται η μαθηματική μοντελοποίηση του προβλήματος, το οποίο διαπραγματεύεται η συγκεκριμένη εργασία.

3.2 Master Problem

Ας πάρουμε για αρχή το παρακάτω γραμμικό πρόβλημα ελαχιστοποίησης, το οποίο για λόγους διευκόλυνσης θα το ονομάσουμε **πρόβλημα(1)**. Η αντικειμενική συνάρτηση είναι:

$$\text{minimize } z_{MP} = \sum_{i \in S} c_i x_i \quad (1)$$

με περιορισμούς,

$$\sum_{i \in S} a_{if} x_i \geq b_f, \quad \forall f \in F \quad (2)$$

$$x_i \geq 0, \quad i \in S \quad (3)$$

Όπου c_i , a_{if} , b_f , x_i ανήκουν στο σύνολο των πραγματικών αριθμών R , για κάθε $i \in S$, όπου $S = \{1, 2, \dots, n\}$, $F = \{1, 2, \dots, m\}$ και n, m είναι ο αριθμός των μεταβλητών του προβλήματος και ο αριθμός των περιορισμών αντίστοιχα. Η βέλτιστη λύση του προβλήματος αυτού συμβολίζεται Z_{MP}^* . Γίνεται η υπόθεση ότι ο αριθμός των μεταβλητών του προβλήματος είναι πολύ μεγαλύτερος από τον αριθμό των περιορισμών. Λόγω αυτού, μετατρέπουμε το πρόβλημα σε ένα μικρότερο πρόβλημα, λαμβάνοντας υπόψη λιγότερες μεταβλητές, δηλαδή έχουμε ένα υποσύνολο του S το S_1 . Το νέο πρόβλημα είναι το restricted master problem (RMP) και διαμορφώνεται ως ακολούθως:

$$\text{minimize } z_{MP} = \sum_{i \in S_1} c_i x_i \quad (4)$$

με περιορισμούς,

$$\sum_{i \in S_1} a_{if} x_i \geq b_f, \quad \forall f \in F \quad (5)$$

$$x_i \geq 0, \quad i \in S_1 \quad (6)$$

Η βέλτιστη λύση στο RMP αποτελεί ένα πάνω όριο για το master problem(MP). Επίσης δεδομένου ότι ο αριθμός των περιορισμών είναι ίδιος και στα δύο προβλήματα, κάθε εφικτή δυική λύση του RMP είναι εφικτή και στο MP.

Ας περάσουμε τώρα στο δικό μας πρόβλημα. Όπως είπαμε σκοπός του προβλήματος είναι η κάλυψη ενός συνόλου διαδρομών S , από ένα σύνολο ιπταμένων I , ικανοποιώντας ταυτόχρονα κάποια ποιοτικά πρότυπα. Πιο συγκεκριμένα, κάθε διαδρομή έχει μια αφετηρία και έναν προορισμό. Τι εννοούμε όμως με τον όρο διαδρομή. Ως διαδρομή ορίζεται ένα ταξίδι από ένα μέρος σε ένα άλλο. Ωστόσο αυτό το ταξίδι μπορεί να μην είναι απευθείας. Ενδιάμεσα μπορεί να έχει και άλλους προορισμούς, μέχρι να φτάσει στον τελικό του προορισμό. Η διάρκεια πτήσης κάθε διαδρομής διαφέρει και μπορεί να είναι και παραπάνω από μια ημέρα. Αυτό οφείλεται στο γεγονός ότι στους ενδιάμεσους προορισμούς, λόγω των πολλών ωρών που μπορεί να ταξιδεύει, θα χρειαστεί να διακόψει την διαδρομή για να ξεκουραστεί το πλήρωμα και να συνεχίσει μετά τον κατάλληλο χρόνο ξεκούρασης. Επίσης πολλές φορές το πλήρωμα χρειάζεται να διανυκτερεύσει σε έναν από τους ενδιάμεσους προορισμούς και να συνεχίσει την πτήση το επόμενο πρωί. Όλα αυτά συμβάλουν στην αύξηση της διάρκειας της ολικής διαδρομής. Το σύνολο των διαδρομών που καλύπτει ο κάθε πιλότος στην διάρκεια ενός μήνα, το ονομάζουμε πρόγραμμα του πιλότου.

Σε κάθε πιλότο αντιστοιχεί ένα διάστημα ωρών πτήσης, που πρέπει να καλύψει. Εάν ο συνολικός αριθμός ωρών που θα πετάξει κάθε ιπτάμενος, ο οποίος προκύπτει ως το άθροισμα της συνολικής διάρκειας πτήσης όλων των διαδρομών του προγράμματός του, είναι έξω από το διάστημα που του αντιστοιχεί (είτε περισσότερες, είτε λιγότερες), τότε για κάθε ώρα εκτός του διαστήματος έχουμε ένα κόστος, το οποίο ονομάζεται balance cost. Το balance cost, το οποίο στην ουσία είναι το κόστος του penalty που αντιστοιχεί στην εταιρεία για τη μη κάλυψη των απαιτούμενων ωρών κάθε ιπταμένου, στην συγκεκριμένη εργασία ορίζεται 50 μονάδες ανά ώρα εκτός του διαστήματος. Επίσης, αν μια διαδρομή μείνει ακάλυπτη, για κάθε ημέρα που η αποστολή αυτή παραμένει ακάλυπτη ορίζεται ένα

κόστος, το κόστος ακάλυπτης διαδρομής. Είναι πολύ σημαντικό η εταιρεία να μπορεί να πραγματοποιήσει όλες τις διαδρομές που έχει προγραμματίσει για ένα χρονικό διάστημα (εδώ για διάρκεια ενός μήνα). Για κάθε διαδρομή, η οποία έχει την κατάλληλη ζήτηση, ώστε να προγραμματιστεί και δεν μπορεί τελικά να πραγματοποιηθεί, το κόστος για την εταιρεία είναι τεράστιο. Στην συγκεκριμένη εργασία το κόστος την ακάλυπτης διαδρομής ορίζεται ως 1.000.000 μονάδες ανά ημέρα που η διαδρομή παραμένει ακάλυπτη. Τα κόστη αυτά ορίζονται ως παράμετροι του προβλήματος, όμως το κόστος ακάλυπτης διαδρομής, όπως παρατηρούμε, είναι συγκριτικά με το balance cost πολύ μεγαλύτερο, δεδομένου ότι άμεσος στόχος του προβλήματος είναι η κάλυψη όσο το δυνατόν περισσότερων διαδρομών και δευτερεύον στόχος είναι η μείωση του κόστους, το οποίο αντιστοιχεί σε κάθε ώρα πτήσης, που ο κάθε πιλότος είναι εκτός του διαστήματος των ωρών, τις οποίες πρέπει να καλύψει.

Επιπλέον, ένα σημαντικό κριτήριο το οποίο πρέπει να ικανοποιηθεί, αφορά την χρονική επικάλυψη των διαδρομών. Κατά την επίλυση του προβλήματος για τον ίδιο πιλότο δημιουργούνται κάποια προγράμματα, κάθε ένα από τα οποία περιέχει το δικό του συνδυασμό διαδρομών. Όλες οι διαδρομές του προγράμματος, που θα επιλεγεί για τον κάθε πιλότο, πρέπει οπωσδήποτε να καλυφθούν. Επομένως, το κάθε πρόγραμμα δεν μπορεί να περιέχει διαδρομές, οι οποίες είναι αλληλεπικαλυπτόμενες χρονικά. Δυο διαδρομές δεν θεωρούνται αλληλεπικαλυπτόμενες χρονικά, όταν η ημέρα επιστροφής της μίας είναι ίδια ή προηγούμενη ημέρα, από την ημέρα αναχώρησης της άλλης διαδρομής. Επίσης όσο αναφορά τους πιλότους, κάποιες ημερομηνίες μπορεί να μην είναι διαθέσιμοι για πτήση, λόγω άδειας, crew rest κ.α. Απόρροια αυτού είναι ότι οι διαδρομές, οι οποίες αντιστοιχούν στις συγκεκριμένες ημερομηνίες, δεν μπορούν να καλυφτούν από πιλότους, οι οποίοι δεν είναι διαθέσιμοι τις ημερομηνίες αυτές.

Παρακάτω θα παρουσιάσουμε την μαθηματική μοντελοποίηση του προβλήματος, αφού πρώτα ορίσουμε κάποια μαθηματικά σύμβολα.

Σύνολα

I : το σύνολο των πιλότων,

S : το σύνολο των διαδρομών,

R_i : το σύνολο των προγραμμάτων για τον ιπτάμενο i ,

Παράμετροι του προβλήματος

b : το κόστος που αντιστοιχεί όταν είναι εκτός του διαστήματος των ωρών πτήσης, δηλαδή το balance cost επί την διαφορά του συνόλου των ωρών από το κάτω όριο ή το πάνω όριο του διαστήματος, ανάλογα με το αν οι ώρες είναι λιγότερες από το κάτω όριο του διαστήματος ή περισσότερες από το πάνω όριο. Αν είναι μέσα στα όρια τότε τίθεται ίσο με μηδέν,

f : το κόστος για τις ακάλυπτες διαδρομές, το οποίο είναι το κόστος ακάλυπτης διαδρομής επί τη διάρκεια που μένει ακάλυπτη η διαδρομή σε ημέρες,

a_{ijs} : δυαδική παράμετρος, η οποία παίρνει την τιμή 1 όταν το πρόγραμμα j του πιλότου i καλύπτει την διαδρομή s , αλλιώς παίρνει την τιμή 0, $i \in I, j \in R_i, s \in S$,

Μεταβλητές απόφασης

x_{ij} : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1 όταν επιλέγεται το πρόγραμμα j του πιλότου i , και 0 διαφορετικά, $i \in I, j \in R_i$,

y_s : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1 όταν η διαδρομή s παραμένει ακάλυπτη, και 0 διαφορετικά, $s \in S$.

Έχοντας ορίσει τα παραπάνω σύμβολα, θα παρουσιάσουμε παρακάτω το μαθηματικό μοντέλο του master problem.

Η αντικειμενική συνάρτηση προκύπτει ως εξής:

$$\text{minimize } z_{MP} = \sum_{i \in I} \sum_{j \in R_i} b x_{ij} + \sum_{s \in S} f y_s \quad (7)$$

με περιορισμούς,

$$\sum_{j \in R_i} x_{ij} = 1, \quad \forall i \in I \quad (8)$$

$$y_s + \sum_{i \in I} \sum_{i \in S_1} a_{ijs} x_{ij} = 1, \quad \forall s \in S \quad (9)$$

$$x_{ij} y_s \text{ binary}, \forall i, j, s \quad (10)$$

Η εξίσωση (7) αποτελεί την αντικειμενική συνάρτηση και ελαχιστοποιεί το κόστος, το οποίο αποτελείται από το κόστος για τις ώρες εκτός του διαστήματος του κάθε ιπταμένου και το κόστος για τις ακάλυπτες διαδρομές. Όπως έχουμε αναφέρει και παραπάνω ο συντελεστής f είναι πάντα πολύ μεγαλύτερος σε σχέση με τον συντελεστή b , δίνοντας έτσι προτεραιότητα στην κάλυψη των διαδρομών. Οι περιορισμοί (8) εξασφαλίζουν ότι για κάθε πιλότο θα επιλεγεί μόνο ένα πρόγραμμα. Ο περιορισμός αυτός περιλαμβάνει και τα προγράμματα τα οποία είναι κενά, δηλαδή αυτά που δεν περιέχουν καμία διαδρομή. Οι περιορισμοί (9) δηλώνουν ότι κάθε διαδρομή s , είτε θα καλυφθεί από ένα ακριβώς πρόγραμμα, είτε θα μείνει ακάλυπτη και η μεταβλητή y_s θα πάρει την τιμή 1. Τέλος οι περιορισμοί (10) υποδηλώνουν ότι οι μεταβλητές απόφασης είναι δυαδικές.

Το παραπάνω μοντέλο χρησιμοποιείται αρκετά στα προβλήματα αερομεταφοράς, τα οποία συνήθως είναι προβλήματα με μεγάλο αριθμό

μεταβλητών. Επίσης, παρόμοια μορφοποίηση χρησιμοποιείται και για προβλήματα με στόλους από οχήματα, τρένα ή πλοία. Σχεδόν πάντα το master problem παρουσιάζει μια βασική παρόμοια δομή για τον προσδιορισμό των διαδρομών των οχημάτων, τρένων ή πλοίων, τα οποία εκτελούν ένα συγκεκριμένο πρόγραμμα, αποσκοπώντας συνήθως σε όσο το δυνατόν μικρότερο συνολικό κόστος. Από την άλλη, πολύπλοκοι περιορισμοί που σχετίζονται με ειδικές απαιτήσεις του προβλήματος, ενσωματώνονται στη μορφοποίηση του προβλήματος column generation. Αυτό έχει ως αποτέλεσμα τη διατήρηση πιο απλής δομής για το master problem και την πιο εύκολη επίλυσή του.

3.3 Column Generation Sub-problem

Ο σκοπός του column generation sub-problem είναι να βρει τους συνδυασμούς προγραμμάτων-ιπταμένων με τον μικρότερο reduce cost. Εάν αυτό το κόστος είναι αρνητικό, τότε υποδεικνύει ότι ο συνδυασμός προγράμματος-ιπταμένου έχει δυνατότητα να βελτιώσει περεταίρω την λύση. Επομένως, όταν βρεθεί εισάγεται στο πρόβλημα, το οποίο λύνεται ξανά και μας δίνει νέα βέλτιστη λύση και νέα δυική λύση. Σύμφωνα με τον Κοζανίδη (2014), εάν δεν βρεθεί συνδυασμός με αρνητικό κόστος, τότε δεν μπορεί να βελτιωθεί περεταίρω η λύση και η διαδικασία αυτή τελειώνει.

Ας επιστρέψουμε τώρα στο γενικό πρόβλημα, δηλαδή στο **πρόβλημα(1)**.

Η εξίσωση που θα μας δώσει το reduce cost μπορεί να γραφτεί ως εξής:

$$c_{RC\ i} = c_i - \sum_{f \in F} a_{if} \pi_f \quad (11)$$

Όπου c_i είναι το κόστος της μεταβλητής i , a_{if} είναι δυική μεταβλητή, η οποία παίρνει τιμή 1 όταν η μεταβλητή i υπάρχει στον περιορισμό f και π_f είναι η δυική τιμή του περιορισμού f . Η μεταβλητή i που θα μας δώσει το μικρότερο C_{RC} θα εισαχθεί στο RMP ως νέα μεταβλητή. Το πρόβλημα θα λυθεί ξανά και η διαδικασία θα συνεχιστεί έως ότου να μην υπάρχει αρνητικό reduce cost , δηλαδή όταν θα ισχύει $C_{RC} \geq 0$, όπου και το πρόβλημα δεν θα μπορεί να βελτιστοποιηθεί παραπάνω. Η βέλτιστη λύση του RMP θα είναι και η βέλτιστη λύση του MP.

Στο πρόβλημα, το οποίο διαπραγματευόμαστε στην συγκεκριμένη εργασία, το column generation sub-problem θα λυθεί με την βοήθεια ενός δικτύου $N=\{A,V\}$. Πριν αναλύσουμε όμως τον τρόπο με τον οποίο θα χρησιμοποιηθεί το δίκτυο, θα πρέπει να ορίσουμε κάποιες παραμέτρους. Έστω sd_s , ld_s είναι η ημερομηνία αναχώρησης και η ημερομηνία προσγείωσης της διαδρομής s . Η διάρκεια της πτήσης (Duration) μετράται σε ημέρες. Ωστόσο, κάθε διαδρομή s έχει και ένα χρόνο πτήσης (Flight time), που μετράται σε ώρες πτήσης. Ας επιστρέψουμε και πάλι στο δίκτυο N . Το δίκτυο περιλαμβάνει έναν κόμβο για κάθε πιλότο, τον οποίο ονομάζουμε κόμβο 0, έναν κόμβο για κάθε διαδρομή, καθώς και έναν τελικό κόμβο, ο οποίος ονομάζεται fictitious και θα συμβολίζεται ως κόμβος E , όπου εκεί τερματίζει κάθε πρόγραμμα. Ο κόμβος fictitious δεν υπάρχει στην πραγματικότητα, δηλαδή είναι φανταστικός κόμβος και τον χρησιμοποιούμε για δική μας ευκολία. Αφού ο κόμβος 0 είναι ξεχωριστός για κάθε πιλότο, καταλαβαίνουμε ότι έχουμε τόσα δίκτυα όσοι είναι και οι πιλότοι.

Μεταξύ των κόμβων του δικτύου καθορίσαμε δυο διαφορετικούς συνδυασμούς. Ο πρώτος συνδυασμός περιλαμβάνει το ζευγάρι κόμβων πιλότου-διαδρομής (I-S), ενώ ο δεύτερος το ζευγάρι κόμβων διαδρομής-διαδρομής (S-S). Για τον συνδυασμό I-S, ορίζουμε ένα σύνολο L_i , το οποίο περιλαμβάνει τις διαδρομές s , που είναι διαθέσιμες να κάνει ο πιλότος i , δηλαδή αυτές που μπορούν να είναι οι επόμενοι συμβατοί κόμβοι, με τους οποίους θα συνδυαστεί ο κόμβος του πιλότου. Επίσης ορίζουμε και ένα σύνολο B_s , το οποίο περιλαμβάνει τους πιλότους i , οι οποίοι μπορούν να αποτελέσουν τον προηγούμενο συμβατό κόμβο της διαδρομής s . Ο κόμβος ενός πιλότου θεωρείται ως προηγούμενος συμβατός κόμβος με τον κόμβο της διαδρομής s , όταν ο κόμβος της διαδρομής s μπορεί να αποτελέσει τον επόμενο συμβατό κόμβο του πιλότου i . Για τον συνδυασμό S-S, ορίζουμε δύο σύνολα το P_s

και το N_s . Για κάθε διαδρομή s , ορίζουμε το σύνολο N_s , το οποίο περιλαμβάνει όλες τις διαθέσιμες διαδρομές, που μπορούν να αποτελέσουν τις επόμενες συμβατές διαδρομές για να συνδυαστούν με την διαδρομή s . Μια διαδρομή k μπορεί να αποτελέσει την επόμενη συμβατή διαδρομή για να συνδυαστεί με την διαδρομή s , όταν $ld_s \leq sd_k$. Επιπλέον, για κάθε διαδρομή s ορίζουμε το σύνολο P_s , το οποίο αποτελείται από τις διαδρομές, οι οποίες μπορούν να αποτελέσουν προηγούμενους συμβατούς κόμβους της διαδρομής s . Μια διαδρομή k μπορεί να αποτελέσει προηγούμενο συμβατό κόμβο, του κόμβου της διαδρομής s , όταν η διαδρομή s είναι επόμενη συμβατή διαδρομή για να συνδυαστεί με τον κόμβο της διαδρομής k .

Σκοπός του column generation problem είναι, λαμβάνοντας υπόψη τους περιορισμούς, να βρει το καλύτερο πρόγραμμα για κάθε ιπτάμενο, συνδυάζοντας τις διαδρομές, το οποίο θα ξεκινάει από τον κόμβο 0 του πιλότου και θα τελειώνει στον fictitious και θα έχει το μικρότερο reduce cost. Λύνοντας το master problem παίρνουμε τις τιμές της δυικής λύσης. Έστω d_i και t_s οι δυικές τιμές των περιορισμών των πιλότων και των διαδρομών αντίστοιχα. Τα σύνολα L_i , B_s , P_s και N_s έχουν οριστεί παραπάνω. Παρακάτω παρουσιάζουμε την μαθηματική μοντελοποίηση του προβλήματος column generation για το πρόβλημά μας.

Μεταβλητές απόφασης

p_i : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται αφορά τον πιλότο i , αλλιώς παίρνει την τιμή 0, $i \in I$,

x_{is} : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται περιλαμβάνει απευθείας σύνδεση του κόμβου 0 του πιλότου i με τον κόμβο της διαδρομής s , αλλιώς παίρνει την τιμή 0, $i \in I$, $s \in L_i$,

x_{sk} : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται περιλαμβάνει απευθείας σύνδεση από τον κόμβο

της διαδρομής s στον κόμβο της διαδρομής k , αλλιώς παίρνει την τιμή 0. Ο κόμβος k μπορεί να είναι είτε κόμβος μιας άλλης διαδρομής είτε ο τελικός κόμβος E (fictitious), $s \in S, k \in N_s \cup \{E\}$,

c_s : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται περιλαμβάνει την διαδρομή s , αλλιώς παίρνει την τιμή 0, $s \in S$.

$$\text{minimize } RC = b - \sum_{i \in I} d_i p_i - \sum_{s \in S} t_s c_s \quad (12)$$

με περιορισμούς,

$$\sum_{i \in I} p_i = 1 \quad (13)$$

$$\sum_{s \in L_i} x_{is} = p_i, \quad \forall i \in I \quad (14)$$

$$\sum_{i \in B_s} x_{is} + \sum_{t \in P_s} x_{ts} = \sum_{k \in N_s \cup \{E\}} x_{sk}, \quad \forall s \in S \quad (15)$$

$$c_s = \sum_{i \in B_s} x_{is} + \sum_{t \in P_s} x_{ts}, \quad \forall s \in S \quad (16)$$

$$p_i, x_{is}, x_{ts}, x_{sk}, c_s \text{ δυικές}(0 \text{ ή } 1), \forall i, s, t, k \quad (17)$$

Η εξίσωση (12) αποτελεί και την αντικειμενική συνάρτηση η οποία χρησιμοποιεί τις δυικές τιμές για να βρει το μικρότερο reduce cost. Οι περιορισμοί (13) εξασφαλίζουν ότι κάθε πρόγραμμα θα ανήκει σε έναν μόνο ιπτάμενο. Οι περιορισμοί (14) δηλώνουν ότι το πρόγραμμα που θα βρεθεί, πρέπει να περιλαμβάνει μια απευθείας σύνδεση μεταξύ του κόμβου του πιλότου με τον κόμβο μιας από τις διαδρομές που μπορούν να είναι οι επόμενες συμβατές. Το σύνολο των περιορισμών (15) δηλώνουν ότι σε έναν κόμβο θα εισέρχεται το πρόβλημα είτε από έναν κόμβο πιλότου, είτε μιας άλλης διαδρομής. Επίσης, από τον κόμβο αυτό θα εξέρχεται είτε προς έναν άλλον κόμβο διαδρομής, είτε προς τον τελικό κόμβο. Οι περιορισμοί (16) εξασφαλίζουν ότι μια διαδρομή καλύπτεται από ένα πρόγραμμα, αν και μόνο αν η διαδρομή αυτή έχει συνδυαστεί με προηγούμενες, δηλαδή μόνο εάν υπάρχει κάποια είσοδος σε αυτή την διαδρομή. Τέλος, οι περιορισμοί (17) δηλώνουν το είδος των μεταβλητών του προβλήματος.

Ωστόσο, αξίζει να σημειωθεί, ότι κατά την επίλυση του όλου προβλήματος με το πρόγραμμα IBM ILOG CPLEX Optimization Studio, το πρόβλημα column generation, το οποίο παρουσιάσαμε παραπάνω, δεν επιλύθηκε ως ξεχωριστό πρόβλημα. Για την επίλυσή του εφαρμόστηκε ο αλγόριθμος συντομότερης διαδρομής για μη κυκλικά δίκτυα (acyclic shortest path algorithm, ASPA).

3.4 Branching

Όταν η λύση του προβλήματος γραμμικής χαλάρωσης περιέχει δεκαδικές τιμές, ο αλγόριθμος προβλέπει την διακλάδωση (branching) του προβλήματος σε άλλα υπο-προβλήματα, προκειμένου να συνεχιστεί η έρευνα για την εύρεση της βέλτιστης ακέραιης λύσης. Όπως και στον αλγόριθμο branch and bound, τα νέα υπο-προβλήματα δημιουργούνται λόγω της προσθήκης περιορισμών, οι οποίοι

εξαλείφουν την δεκαδική λύση. Για την δημιουργία αυτών των υπο-προβλημάτων, απαραίτητη είναι η επιλογή μιας μεταβλητής απόφασης (decision variable) με δεκαδική τιμή. Όταν μια τέτοια μεταβλητή βρεθεί το πρόβλημα διαχωρίζεται σε δύο επιμέρους υπο-προβλήματα, όπου στο ένα η τιμή της μεταβλητής αυτής θέτεται ίση με μηδέν και στο άλλο θέτεται ίση με την μονάδα.

Αποδεικνύεται ότι εάν η βέλτιστη λύση του προβλήματος γραμμικής χαλάρωσης περιέχει μια ή και περισσότερες μεταβλητές απόφασης με δεκαδική τιμή, τότε υπάρχει τουλάχιστον ένας περιορισμός για τους ιπταμένους και ένας για τις διαδρομές, τέτοιοι ώστε το άθροισμα των μεταβλητών, οι οποίες εμφανίζονται και στους δύο περιορισμούς, να προκύπτει δεκαδική τιμή.

Λαμβάνοντας υπόψη τα παραπάνω, προκύπτει ένα εύκολος τρόπος για να γίνει η επιλογή της μεταβλητής. Όταν η βέλτιστη λύση του προβλήματος γραμμικής χαλάρωσης περιέχει δεκαδικές τιμές, μπορούν να βρεθούν τουλάχιστον δυο περιορισμοί που να ικανοποιούν τα παραπάνω. Εάν η λύση περιλαμβάνει μόνο μια μεταβλητή με δεκαδική τιμή, τότε η επιλογή της είναι προφανής. Στην περίπτωση όμως, που υπάρχουν πολλές μεταβλητές με δεκαδική τιμή, τότε εμείς ορίσαμε να επιλέγεται η μεταβλητή, η οποία έχει την μεγαλύτερη δεκαδική τιμή. Έχοντας αυτή την μεταβλητή εφαρμόζεται η μεθοδολογία της διακλάδωσης. Στο αριστερό υπο-πρόβλημα, η τιμή της μεταβλητής αυτής θέτεται ίση με το μηδέν, ενώ στο δεξί υπο-πρόβλημα ίση με την μονάδα. Αυτό σημαίνει ότι για τον ιπτάμενο, με τον οποίον συνδέεται η μεταβλητή, είτε θα επιλεγεί ο συνδυασμός διαδρομών που υποδηλώνει αυτή η μεταβλητή (όταν η τιμή της είναι 1), δηλαδή το πρόγραμμα που αντιπροσωπεύει, είτε ότι η μεταβλητή αυτή δεν θα συμμετέχει και δεν θα επηρεάσει την λύση (όταν η τιμή της είναι 0). Το πλεονέκτημα της εφαρμογής αυτού είναι ότι ο αριθμός των περιορισμών παραμένει ο ίδιος και δεν χρειάζεται η προσθήκη νέων. Επομένως δεν χρειάζεται να ασχοληθούμε με παραπάνω δυικές μεταβλητές, χωρίς να χρειάζεται.

Στο αριστερό υπο-πρόβλημα η μεταβλητή, η οποία έχει επιλεχθεί, θέτεται ίση με το μηδέν και στην ουσία διαγράφεται από το πρόβλημα. Ένας κατάλληλος περιορισμός εισάγεται στο column generation sub-problem, ώστε η μεταβλητή αυτή να μην δημιουργηθεί ξανά. Λύνουμε ξανά το πρόβλημα και οι νέες μεταβλητές, οι

οποίες θα παραχθούν, θα διαφέρουν από αυτήν που διαγράφηκε, το οποίο βοηθάει στην εύρεση καλύτερης λύσης.

Στο δεξί υπο-πρόβλημα η μεταβλητή αυτή θέτεται ίση με 1. Αυτό, όπως είπαμε, δεν γίνεται με την προσθήκη νέων περιορισμών, αλλά με την κατάλληλη μορφοποίηση των ήδη υπαρχόντων. Όλες οι άλλες μεταβλητές, οι οποίες υπάρχουν στους περιορισμούς στους οποίους εντοπίζεται και η επιλεχθείσα μεταβλητή, διαγράφονται από το πρόβλημα και με αυτόν τον τρόπο προκύπτει το αποτέλεσμα που θέλουμε, δηλαδή να ισούται η μεταβλητή που επιλέξαμε με την μονάδα. Κατάλληλοι περιορισμοί εισάγονται στο column generation sub-problem, προκειμένου οι μεταβλητές που διαγράφηκαν να μην ξαναδημιουργηθούν, καθώς οι περιορισμοί, με τους οποίους σχετίζεται η επιλεχθείσα μεταβλητή, έχουν ικανοποιηθεί. Στην συνέχεια λύνουμε ξανά το πρόβλημα, το οποίο προκύπτει έχοντας κάνει την παραπάνω διαδικασία.

Συγκρίνοντας τις λύσεις που προκύπτουν από αυτά τα δύο προβλήματα, επιλέγουμε αυτήν, η οποία μας δίνει την καλύτερη λύση. Έχοντας επιλέξει τον κλάδο που δίνει την καλύτερη λύση, το πρόβλημα αναθεωρείται και το νέο πρόβλημα γίνεται το πρόβλημα του κλάδου αυτού. Εάν η λύση περιέχει δεκαδικές τιμές, η διαδικασία αυτή επαναλαμβάνεται, έως ότου να βρεθεί η βέλτιστη ακέραια λύση.

ΚΕΦΑΛΑΙΟ 4

ΜΕΘΟΔΟΛΟΓΙΑ ΕΠΙΛΥΣΗΣ

4.1 Εισαγωγή

Στο κεφάλαιο 3 παρουσιάσαμε την μαθηματική μοντελοποίηση, η οποία διέπει το πρόβλημά μας. Ωστόσο, για την επίλυση του προβλήματος με τον πιο αποτελεσματικό τρόπο, πρέπει να λάβουμε υπόψη μας και κάποιες λεπτομέρειες, οι οποίες μπορούν να επηρεάσουν σημαντικά την λύση και δεν είναι εμφανείς στην μαθηματική μοντελοποίηση. Εδώ αξίζει να σημειωθεί ότι πολλές φορές, αρκετές από αυτές τις λεπτομέρειες, αποτελούν και παραμέτρους του προβλήματος, τις οποίες καλούμαστε να αποφασίσουμε συνήθως πριν την επίλυση του προβλήματος.

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τον τρόπο επίλυσης του προβλήματος αλγοριθμικά, λαμβάνοντας υπόψη και τους διάφορες παράγοντες, οι οποίοι μπορούν να επηρεάσουν την λύση. Επιπλέον θα παρουσιάσουμε και ένα αριθμητικό παράδειγμα, ώστε να γίνει πιο κατανοητός ο τρόπος επίλυσης του προβλήματος.

4.2 Αρχικοποίηση του RMP

Για την εφαρμογή της μεθόδου column generation είναι αναγκαίο να υπάρξει μια αρχική εφικτή λύση. Η λύση αυτή θα προκύψει από την λύση του RMP. Στην συγκεκριμένη εργασία, αλλά και γενικότερα στην βιβλιογραφία (M. E. Lübbecke και J. Desrosiers, 2002), η μεθοδολογία που ακολουθείται για την επίλυση του προβλήματος, ως αρχικό πρόβλημα, δηλαδή ως RMP, περιέχει το πρόβλημα το

οποίο σχετίζεται με τις μεταβλητές που αντιστοιχούν στο μεγαλύτερο κόστος. Αυτές οι μεταβλητές, οι οποίες είναι οι μεταβλητές που σχετίζονται με την κάλυψη των διαδρομών αλλά και οι μεταβλητές των προγραμμάτων που δεν περιέχουν καμία διαδρομή, δηλαδή των κενών προγραμμάτων, εάν πάρουν την τιμή 1 (οι αντίστοιχες διαδρομές που αντιπροσωπεύουν να μείνουν ακάλυπτες και οι πιλότοι, στους οποίους αντιστοιχούν τα κενά προγράμματα να μην τους ανατεθεί κάποια διαδρομή με συνέπεια να μην πετάξουν καθόλου), προσθέτουν ένα μεγάλο κόστος στην αντικειμενική συνάρτηση. Εδώ πρέπει να σημειωθεί ότι πολλές φορές κατά την επίλυση αυτού του προβλήματος, λόγω του μεγάλου συντελεστή που έχουν οι μεταβλητές της κάλυψης, μπορούν να επηρεάσουν και να κάνουν σφιχτά τα όρια των μεταβλητών του δυικού προβλήματος, με αποτέλεσμα για να βρεθεί μια βέλτιστη λύση να χρειάζεται κάποιες διαδρομές να μείνουν ακάλυπτες. Άλλοι τρόποι για να γίνει η αρχικοποίηση και να βρεθεί μια αρχική βέλτιστη λύση, είναι να υπάρχει ήδη μια γνωστή και αποδεκτή λύση για το πρόβλημα. Το αρνητικό μέρος αυτού είναι ότι δεν υπάρχουν αρχικές λύσεις για πολλά προβλήματα και ειδικά για τα δύσκολα προβλήματα.

Πολλές φορές μπορεί να βρεθεί στη βιβλιογραφία και μια μη ολοκληρωμένη λύση για το πρόβλημα. Χρησιμοποιώντας αυτήν όμως, δεν είναι σίγουρο ότι θα καταλήξει το πρόβλημα σε μια ολοκληρωμένη λύση, ή ακόμη και να καταλήξει ενδεχομένως κάποιες από τις διαδρομές, οι οποίες θα μπορούσαν να καλυφθούν, να αναγκαστούν να παραμείνουν ακάλυπτες. Οι. Vanderbeck, (1994) και M. E. Lübbecke και J. Desrosiers, (2002), προτείνουν να ρυθμίζεται το κόστος των μεταβλητών των ακάλυπτων διαδρομών, έτσι ώστε να δίνουν όσο το δυνατόν καλύτερες δυικές τιμές. Αυτό στην συγκεκριμένη εργασία δεν μπορεί να γίνει, καθώς το κόστος της ακάλυπτης διαδρομής θέλουμε να είναι πολύ μεγαλύτερο σε σχέση με τα υπόλοιπα κόστη, προκειμένου να τηρούνται οι προτεραιότητες που θέσαμε στην αντικειμενική συνάρτηση.

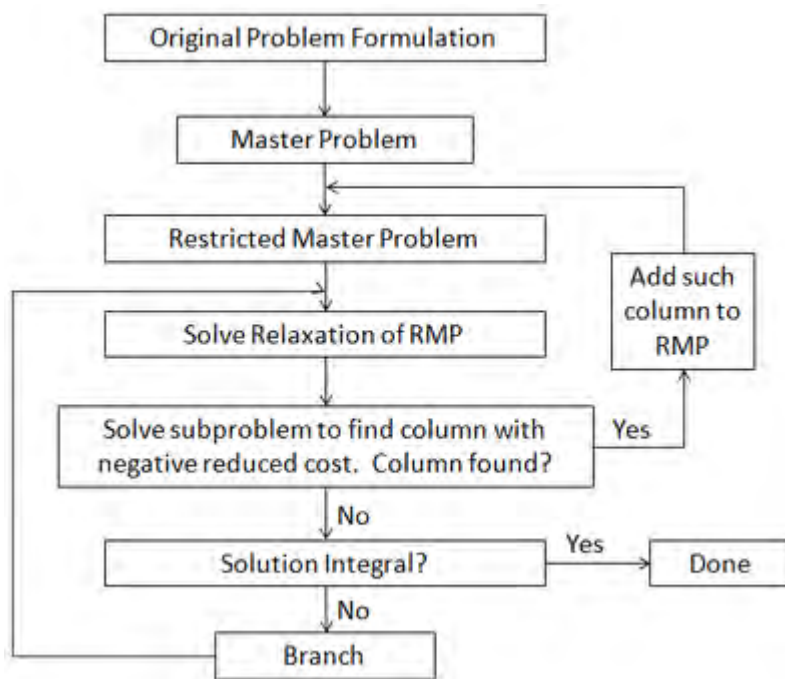
Μια αρχική λύση σε ένα πρόβλημα αποτελεί το πάνω όριο. Αυτό σημαίνει ότι το πρόβλημα δεν μπορεί να έχει χειρότερη λύση από αυτήν. Ωστόσο, χρησιμοποιώντας λύσεις από την βιβλιογραφία έχει αποδειχθεί ότι δεν είναι ότι καλύτερο. Προσθέτοντας μεταβλητές από διάφορες λύσεις που συναντώνται στην βιβλιογραφία, θα έχει ως αποτέλεσμα ο αλγόριθμος του column generation να κάνει

λίγες επαναλήψεις, καθιστώντας τον ουσιαστικά άχρηστο, μη εξυπηρετώντας τον σκοπό, για τον οποίο έχει δημιουργηθεί.

4.3 Επίλυση του master LP relaxation

Όπως έχουμε αναφέρει και στην παράγραφο 3.4, κάθε κόμβος του δέντρου αντιστοιχεί σε ένα ξεχωριστό master problem, το οποίο συνοδεύεται από το αντίστοιχο column generation sub-problem. Οι εξισώσεις, οι οποίες περιγράφουν τα προβλήματα αυτά, είναι αυτές που αναλύθηκαν στο κεφάλαιο 3, με την μόνη διαφορά ότι σε κάθε κόμβο αλλάζουν οι περιορισμοί, ως αποτέλεσμα του branching. Ο συνολικός αριθμός των περιορισμών όμως, παραμένει ο ίδιος σε όλα τα προβλήματα. Για να προκύψει η βέλτιστη λύση του κάθε προβλήματος, το οποίο αντιστοιχεί σε κάθε κόμβο του δέντρου, πρέπει να ακολουθείται η σειρά των βημάτων που δείχνει το σχήμα 4.1, το οποίο φαίνεται παρακάτω.

(Πηγή: Akella, M.; S. Gupta; A. Sarkar. "Branch and Price: Column Generation for Solving Huge Integer Programs" (PDF).)



Σχήμα 4.1: Βήματα για την επίλυση του προβλήματος.

Το πρόβλημα θα λυθεί με την βοήθεια του λογισμικού IBM ILOG CPLEX 12.5.1 (2013). Το λογισμικό αυτό παρέχει διάφορες επιλογές για την λύση του προβλήματος. Οι επιλογές αυτές είναι έξι (6) σε αριθμό και πιο συγκεκριμένα είναι οι:

- optimize
- primal simplex
- dual simplex
- network optimizer
- barrier optimizer
- mixed integer optimizer.

Η επιλογή optimize θεωρείται ως η default επιλογή για την επίλυση του προβλήματος. Σε αυτή την επιλογή το πρόβλημα επιλέγει από μόνο του ποια από όλες τις μεθόδους θα επιλέξει. Οι αλγόριθμοι simplex και barrier χρησιμοποιούνται όταν θέλουμε να πάρουμε και τις τιμές του δυικού προβλήματος. Πιο πολύ χρησιμοποιείται η επιλογή primal simplex, η οποία είναι η πιο κοντινή στην μέθοδο του column generation. Το μεγαλύτερο πλεονέκτημα αυτής της μεθόδου είναι ότι όταν μπαίνουν καινούργιες μεταβλητές στο πρόβλημα η λύση που είχε δώσει πιο πριν παραμένει εφικτή, πράγμα το οποίο βοηθάει πολύ στο να μειωθεί ο χρόνος για την εκ νέου βελτιστοποίηση του προβλήματος.

Η μέθοδος που χρησιμοποιεί τον αλγόριθμο dual simplex δεν μπορεί να αρχίσει έχοντας μια προηγούμενη λύση προερχόμενη από την μέθοδο primal, ωστόσο είναι κατάλληλη για εκφυλισμένα προβλήματα.

Ο αλγόριθμος barrier χρησιμοποιείται κατά κόρων τα τελευταία χρόνια για την επίλυση μεγάλων προβλημάτων ανάθεσης του προσωπικού σε μεγάλες αεροπορικές εταιρίες. Το μειονέκτημα αυτού του αλγορίθμου είναι ότι δεν μπορεί να ξεκινήσει έχοντας μια αρχική λύση, καθώς και ότι ο αλγόριθμος αυτός δεν είναι κατάλληλος για προβλήματα που έχουν πολλές μεταβλητές σε σχέση με τον αριθμό των περιορισμών.

4.4 Αριθμός μεταβλητών που εισάγονται ανά επανάληψη

Σκοπός του column generation sub-problem είναι η εύρεση μιας τουλάχιστον μεταβλητής με αρνητικό reduce cost, η οποία θα εισαχθεί στο αρχικό πρόβλημα. Η μεταβλητή αυτή, κατά κύριο λόγο, είναι η μεταβλητή με το μικρότερο αρνητικό reduce cost. Ωστόσο, πολλές φορές, είναι καλό να εισάγονται παραπάνω από μια μεταβλητές ανά επανάληψη, διότι μπορεί σε μια επανάληψη μια μεταβλητή x να έχει καλύτερο reduce cost από μια μεταβλητή z , όμως στο τελικό πρόβλημα η μεταβλητή z να δίνει καλύτερο αποτέλεσμα από την μεταβλητή x . Επομένως, εάν επιλεγεί να εισάγεται μια μεταβλητή ανά επανάληψη, η λύση που θα προκύψει μπορεί να είναι

χειρότερη σε σχέση με την τελική λύση που θα προέκυπτε, εάν είχαν εισαχθεί παραπάνω μεταβλητές.

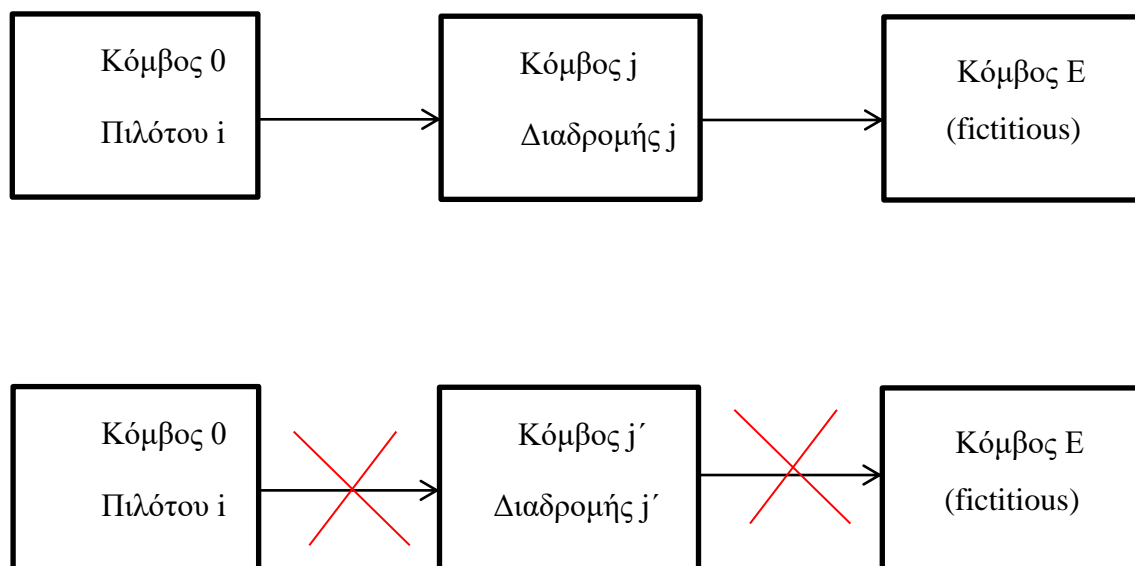
Από την άλλη δεν μπορούν να εισαχθούν και όλες οι μεταβλητές που θα δημιουργηθούν σε μια επανάληψη, καθώς δεν βελτιώνουν την λύση όλες και πολλές μεταβλητές είναι παρόμοιες με κάποιες άλλες. Ωστόσο, εισάγοντας παραπάνω από μια μεταβλητές, στις περισσότερες περιπτώσεις η λύση του προβλήματος που θα προκύψει θα είναι καλύτερη. Εδώ πρέπει να σημειώσουμε, ότι είτε βάλουμε μια μεταβλητή, είτε περισσότερες η τιμή των δυικών μεταβλητών θα μεταβληθεί.

Η αρνητική πλευρά στο να εισαχθούν περισσότερες της μιας μεταβλητές είναι ο μεγαλύτερος χρόνος επίλυσης που θα προκύψει. Σε κάθε κόμβο του δικτύου που δημιουργείται για κάθε πιλότο, αποθηκεύονται παραπάνω από μια τιμές για κάθε μεταβλητή. Όταν οι μεταβλητές είναι παραπάνω από μια, τότε οι τιμές που την αντιπροσωπεύουν αποτελούν την ετικέτα (label) της. Οι τιμές αυτές θα βοηθήσουν στην επίλυση του column generation sub-problem, όπως θα δούμε παρακάτω.

4.5 Επίλυση του column generation sub-problem

Για την επίλυση του Column generation sub-problem δεν θα χρησιμοποιηθεί το λογισμικό IBM ILOG CPLEX 12.5.1. Λόγω του ότι τα δίκτυα που έχουμε είναι μη κυκλικά, για την επίλυση αυτού του προβλήματος θα χρησιμοποιήσουμε τον αλγόριθμο ASPA. Ως απόσταση μεταξύ των κόμβων του δικτύου ορίζεται το reduce cost. Αρχικά θα πρέπει να δώσουμε κάποιες τιμές στα στοιχεία, τα οποία απαρτίζουν τις ετικέτες των κόμβων. Ως ετικέτα ορίζεται το σύνολο των στοιχείων, τα οποία περιγράφουν κάθε κόμβο. Αυτά τα στοιχεία είναι το reduce cost, το Previous, το Previous Path, το Flight Hours και το Sum duals. Τις αρχικές τιμές θα τις πάρουν δημιουργώντας για κάθε κόμβο ένα shortest path, το οποίο ξεκινάει από τον κόμβο 0, δηλαδή του πιλότου, πάει στον κόμβο της διαδρομής και καταλήγει στον κόμβο fictitious, όπως φαίνεται και στο σχήμα 4.2. Εδώ πρέπει να σημειώσουμε ότι αυτό γίνεται μόνο για τους κόμβους που είναι συμβατοί να συνδυαστούν με τον πιλότο.

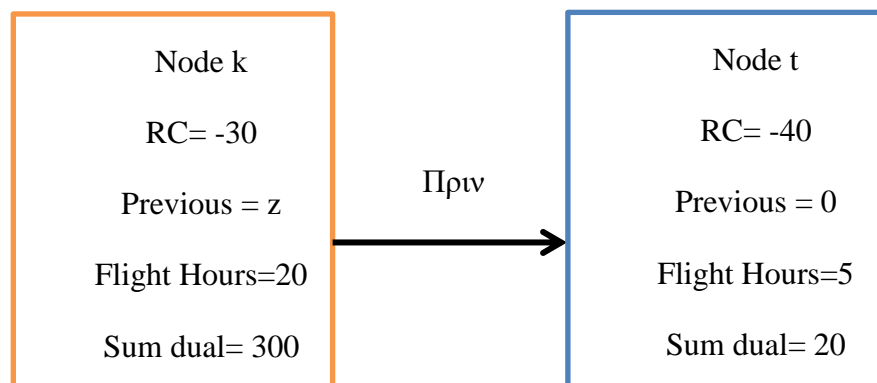
Αν ένας κόμβος, παραδείγματος χάρι ο j' , δεν είναι συμβατός για τον δεδομένο πιλότο δεν θα αρχικοποιηθεί, διότι δεν μπορεί να συμμετέχει στην εύρεση του προγράμματος του πιλότου αυτού. Το *reduce cost* είναι αυτό που καθορίζει εάν δυο εφικτοί κόμβοι θα συνδυαστούν ή όχι. Το πώς γίνεται αυτό θα το περιγράψουμε στο παράδειγμα παρακάτω. Το *Previous*, κάθε ετικέτας του κόμβου, δείχνει τον προηγούμενο κόμβο με τον οποίο συνδυάζεται ο υπό εξέταση κόμβος. Το *Previous Path* χρησιμοποιείται στην περίπτωση που στο πρόβλημα εισάγεται παραπάνω από μια μεταβλητές σε κάθε επανάληψη, για κάθε πιλότο. Όσα είναι τα *Path*, τόσες είναι και οι μεταβλητές που εισάγονται. Το *Previous Path* δείχνει από ποιο *Path*, του προηγούμενου κόμβου με τον οποίο έχει συνδυαστεί, έρχεται η συντομότερη διαδρομή για να συνδυαστεί με το υπό εξέταση *Path* του κόμβου αυτού. Το *Flight hours* κάθε κόμβου περιέχει τις συνολικές ώρες πτήσης των διαδρομών που έχουν συνδυαστεί μέχρι και τον κόμβο αυτό, στο πρόγραμμα που εξετάζεται. Το *Sum duals* του κάθε κόμβου περιλαμβάνει το άθροισμα των δυικών τιμών των κόμβων που έχουν συνδυαστεί μέχρι και τον κόμβο αυτό. Κάθε φορά που μια νέα διαδρομή, δηλαδή ένας νέος κόμβος, συνδέεται στο πρόγραμμα, δηλαδή στο *shortest path*, οι παραπάνω τιμές του κόμβου αυτού αναθεωρούνται.



Σχήμα 4.2: Αρχικοποίηση των τιμών των κόμβων (πάνω είναι η αρχικοποίηση για έναν συμβατό κόμβο, κάτω ο κόμβος j' δεν είναι συμβατός).

Παρακάτω θα δείξουμε ένα παράδειγμα, ώστε αυτά που περιγράψαμε να γίνουν πιο κατανοητά. Το παράδειγμα φαίνεται στο σχήμα 4.3 και στο σχήμα 4.4. Έχουμε δυο κόμβους, τους k και t, οι οποίοι είναι εφικτό να συνδυαστούν. Στο παράδειγμα αυτό εισάγεται μόνο μια μεταβλητή ανά επανάληψη, οπότε δεν χρησιμοποιείται το Previous Path. Το shortest path φτάνει μέχρι τον κόμβο k και θέλουμε να δούμε εάν συνδυάζοντας τους δύο αυτούς κόμβους προκύπτει καλύτερο shortest path. Έστω ότι το shortest path έχει φτάσει μέχρι και τον κόμβο k με reduce cost -30, συνολικό αριθμό ωρών πτήσης 20 και με άθροισμα όλων των δυικών τιμών των κόμβων που έχουν συνδυαστεί, σε αυτό το shortest path, 300. Το Previous μας δείχνει ότι στο πρόγραμμα που δημιουργείται, αμέσως προηγούμενος κόμβος του k είναι ο κόμβος z.

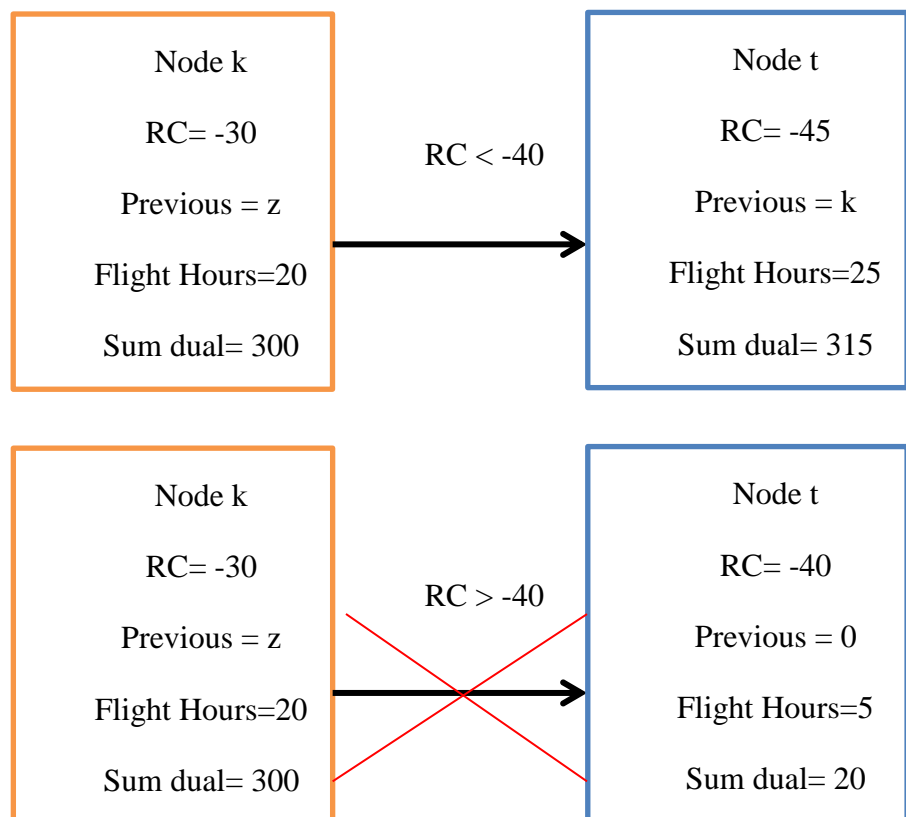
Οι τιμές της ετικέτας του κόμβου t προέρχονται από την αρχικοποίηση του κόμβου. Το reduce cost είναι -40, οι ώρες πτήσης είναι 5, οι οποίες είναι στην ουσία οι ώρες της διαδρομής, καθώς όπως έχουμε πει και παραπάνω για την αρχικοποίηση δημιουργείται ένα shortest path που ξεκινάει από τον κόμβο 0 του πιλότου, πηγαίνει στον κόμβο της διαδρομής και καταλήγει στον fictitious. Το άθροισμα των δυικών τιμών είναι 20 και είναι το άθροισμα της δυικής τιμής του πιλότου και της διαδρομής. Το Previous είναι το 0, δηλαδή ο κόμβος του πιλότου.



Σχήμα 4.3: Ετικέτες των κόμβων πριν τον έλεγχο για συνδυασμό.

Για να ελέγξουμε εάν ο συνδυασμός αυτών των κόμβων θα οδηγήσει σε καλύτερο shortest path, θα πρέπει να ελέγξουμε το reduce cost που προκύπτει από τον συνδυασμό. Εάν το reduce cost είναι μικρότερο του -40, τότε ο συνδυασμός αυτός μας δίνει καλύτερο αποτέλεσμα σε σχέση με το -40 που είχε ο συνδυασμός το κόμβου t με τον κόμβο 0 του πιλότου. Έτσι αμέσως προηγούμενος κόμβος του κόμβου t θα γίνει ο κόμβος k, αναθεωρώντας τις τιμές όλων των στοιχείων της ετικέτας του κόμβου t. Αντίθετα, εάν ο συνδυασμός οδηγήσει σε reduce cost μεγαλύτερο του -40, τότε δεν πρέπει να συνδυαστεί ο κόμβος k με τον κόμβο t, καθώς ο κόμβος t συνδυασμένος με τον αρχικό κόμβο του πιλότου έχει μικρότερο αρνητικό κόστος. Επομένως οι τιμές των στοιχείων της ετικέτας του κόμβου t δεν θα αλλάξουν. Όλα αυτά φαίνονται και στο σχήμα 4.4.

Αυτή η διαδικασία επαναλαμβάνεται, ελέγχοντας τον κόμβο k με όλους τους κόμβους, οι οποίοι είναι συμβατοί να συνδυαστούν μαζί του και βρίσκονται δεξιά του στο δίκτυο, αναθεωρώντας τις ετικέτες τους, όπου αυτό χρειάζεται.

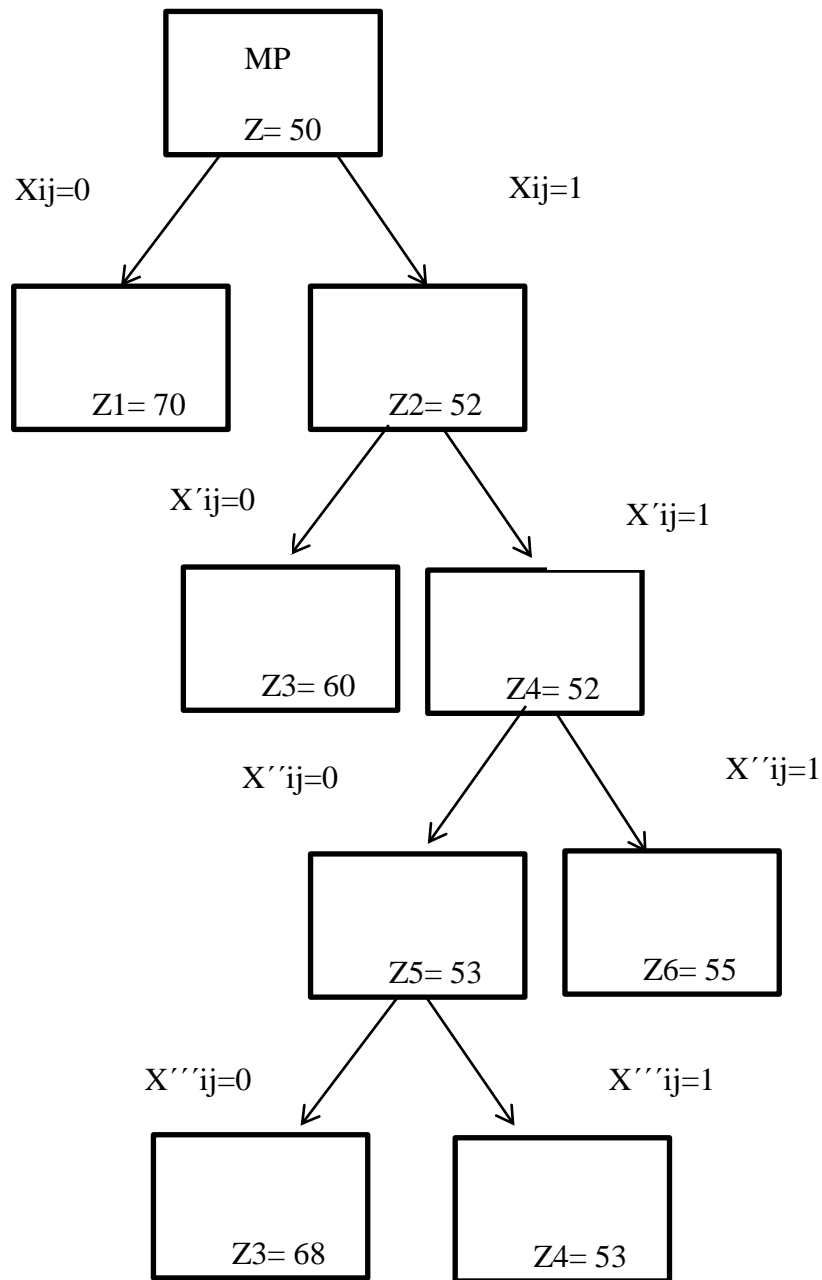


Σχήμα 4.4: Τιμές κόμβου t μετά τον έλεγχο.

4.6 Εφαρμογή του βήματος Branching

Αφού ολοκληρωθεί και η επίλυση του column generation sub-problem, το τελευταίο βήμα για την επίλυση του προβλήματος είναι το δέντρο διακλάδωσης (branching tree). Όπως έχουμε αναφέρει και στην ενότητα 3.4, σε κάθε κλάδο έχουμε και ένα διαφορετικό πρόβλημα, αλλά με τον ίδιο αριθμό περιορισμών. Το πρώτο πρόβλημα που λύνεται, δηλαδή το RMP, αποτελεί και το κάτω όριο για όλους τους κλάδους του προβλήματος. Επομένως, σε κανένα κλάδο δεν μπορούμε να βρούμε καλύτερη τιμή αντικειμενικής συνάρτησης από αυτήν. Το καλύτερο που μπορούμε να βρούμε είναι ίδια τιμή αντικειμενικής συνάρτησης. Σε αυτή τη λύση όμως, παρόλο που είναι η καλύτερη, οι τιμές των μεταβλητών απόφασης δεν είναι ακέραιες, οπότε δεν μπορούμε να τη θεωρήσουμε ως τελική βέλτιστη λύση.

Προκειμένου να προχωρήσουμε από έναν κλάδο στον πιο κάτω θα πρέπει να βρούμε τη μεταβλητή με τη μεγαλύτερη δεκαδική τιμή. Αφού βρεθεί η τιμή αυτή τότε πάμε στον επόμενο κλάδο έχοντας δυο προβλήματα. Ένα στο οποίο η τιμή της μεταβλητής αυτής είναι 0 και ένα στο οποίο η μεταβλητή παίρνει την τιμή 1. Εδώ πρέπει να σημειώσουμε ότι ακολουθούμε τους κλάδους που μας δίνουν την καλύτερη τιμή στην αντικειμενική συνάρτηση. Ένα παράδειγμα με τυχαίες τιμές φαίνεται στο σχήμα 4.5.



Σχήμα 4.5: Δέντρο διακλάδωσης (Branching tree)

Εκτενέστερα έχει αναλυθεί η διαδικασία του branching στην ενότητα 3.4. Στην συγκεκριμένη εργασία, λόγω περιορισμένου χρόνου για την ανάπτυξη και την υλοποίηση της εργασίας και του κώδικα, τον οποίο απαιτούσε, ελέγχεται μόνο η περίπτωση, όπου η τιμή της μεταβλητής παίρνει την τιμή 1. Πολλά παραδείγματα

στην βιβλιογραφία έχουν αποδείξει ότι παίρνοντας μόνο την τιμή 1 οδηγούμαστε στη βέλτιστη ή σε πολύ κοντινή, σε αυτή, προσέγγιση.

4.7 Επίλυση αριθμητικού παραδείγματος

Παρακάτω θα αναλύσουμε βήμα προς βήμα την επίλυση ενός αριθμητικού παραδείγματος $N=20$ πιλότων και $R=60$ διαδρομών. Σε κάθε επανάληψη του column generation επιλέγεται να εισάγονται 3 μεταβλητές για κάθε πιλότο, επομένως τα path θα είναι 3. Το κόστος για κάθε διαδρομή που μένει ακάλυπτη το θέτουμε 1000000 μονάδες, ενώ το κόστος για τις ώρες εκτός του διαστήματος των ωρών πτήσης, τις οποίες πρέπει να καλύψει κάθε ιπτάμενος το θέτουμε 50 μονάδες. Όλα αυτά φαίνονται και στην εικόνα 4.1 Αρχικά πρέπει να διευκρινίσουμε ότι οι τιμές του διαστήματος των ωρών πτήσης κάθε πιλότου, της διάρκειας κάθε διαδρομής, της ημέρας αναχώρησης και προσγείωσης της κάθε διαδρομής, καθώς και των συμβατών διαδρομών για κάθε πιλότο, αλλά και για κάθε διαδρομή με τις οποίες μπορούν να συνδυαστούν, προκύπτουν τυχαία με μια γεννήτρια τυχαίων αριθμών. Επομένως, το πρόβλημα αυτό δεν αντιστοιχεί σε ένα πραγματικό πρόβλημα μιας πραγματικής αεροπορικής εταιρίας, το οποίο ίσως έχει επιλυθεί, αλλά αποτελεί ένα καινούργιο άγνωστο μέχρι στιγμής πρόβλημα. Το πρόβλημα θα μορφοποιηθεί και θα λυθεί μέσω του λογισμικού IBM ILOG CPLEX 12.5.1.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include <ilcplex/ilocplex.h>

#define N 20 // number of crew members
#define R 60 // number of routes
#define paths 3 // number of paths
#define UndercoverCost 1000000 // the cost for not covering a route
#define BalanceCost 50 // the cost for not balanced

```

Εικόνα 4.1: Στοιχεία προβλήματος.

Οι εξισώσεις που περιγράφηκαν στο κεφάλαιο 3, μορφοποιούνται με την μορφή συναρτήσεων στον κώδικα που αναπτύχθηκε. Αρχικά θα πρέπει να οριστεί το RMP. Το RMP θα περιλαμβάνει τις μεταβλητές, οι οποίες θα αντιπροσωπεύουν την κάλυψη των διαδρομών (όταν πάρουν τιμή 1 η αντίστοιχη διαδρομή δεν καλύπτεται), καθώς και τις μεταβλητές, οι οποίες αντιστοιχούν στα κενά προγράμματα των πιλότων. Η αντικειμενική συνάρτηση αποτελείται από αυτές τις μεταβλητές, ο συντελεστής των οποίων είναι το κόστος τους. Στις εικόνες 4.2 και 4.3 φαίνονται οι συναρτήσεις που αναπτύχθηκαν στον κώδικα, με τις οποίες δημιουργείται το RMP, ενώ στην εικόνα 4.4 φαίνεται το πρόβλημα με την μορφή εξισώσεων. Οι πρώτοι 20 περιορισμοί αφορούν τους πιλότους και οι υπόλοιποι τις διαδρομές.

```

int initial_variables(CPXENVptr Env, CPXLPptr Lp)
{
    int i, status;

    MemCType(N+R);
    MemForNewCol(N+R);

    for (i=0; i<=R-1; i++){
        SetBoundObj(i, 0 , CPX_INFBOUND, UndercoverCost*DUR[i]);
    }

    for (i=0; i<=N-1; i++){
        SetBoundObj(R+i, 0 , CPX_INFBOUND, BalanceCost*LHRS[i]);
    }
    status = NewCols( Env, Lp, R+N, 'C');
    if (status != 0){
        print_error("NewCols");
        return -1;
    }
    FreeMemCType();
    FreeMemForNewCol();
    return 0;
}

```

Εικόνα 4.2: Δημιουργία μεταβλητών για το RMP

```

int initial_constrains(CPXENVptr Env, CPXLPptr Lp)
{
    int i, status , count, nzcnt;
    rmatbeg = (int*) malloc((R+N)*sizeof(int));
    MemForIndVal(N+R);
    rmatbeg[0]=0;

    for (i=0; i<=N-1; i++){
        count=0;
        nzcnt=0;
        SetMatindMatval( count,i+R, 1);
        (nzcnt++);
        status=AddRows(NULL, Env, Lp, nzcnt, 1, 'E');
        if (status != 0){
            print_error("AddRows");
            return -1;
        }
        (rmatbeg[i+1]=0);
    }

    for (i=0; i<=R-1; i++){
        count=0;
        nzcnt=0;
        SetMatindMatval( count, i, 1);
        (nzcnt++);
        status=AddRows(NULL, Env, Lp, nzcnt, 1, 'E');
        if (status != 0){
            print_error("AddRows1");
            return -1;
        }
        (rmatbeg[i+1]=0);
    }
    FreeMemForIndVal();
    return 0;
}

```

Εικόνα 4.3: Δημιουργία περιορισμών του RMP.

Minimize

obj: 3000000 x1 + 3000000 x2 + 1000000 x3 + 1000000 x4 + 4000000 x5
+ 4000000 x6 + 1000000 x7 + 2000000 x8 + 4000000 x9 + 2000000 x10
+ 3000000 x11 + 4000000 x12 + 1000000 x13 + 2000000 x14 + 4000000 x15
+ 3000000 x16 + 1000000 x17 + 3000000 x18 + 2000000 x19 + 3000000 x20
+ 2000000 x21 + 2000000 x22 + 2000000 x23 + 2000000 x24 + 1000000 x25
+ 1000000 x26 + 2000000 x27 + 4000000 x28 + 2000000 x29 + 1000000 x30
+ 1000000 x31 + 2000000 x32 + 1000000 x33 + 4000000 x34 + 1000000 x35
+ 3000000 x36 + 2000000 x37 + 2000000 x38 + 2000000 x39 + 2000000 x40
+ 4000000 x41 + 2000000 x42 + 1000000 x43 + 4000000 x44 + 4000000 x45
+ 3000000 x46 + 3000000 x47 + 3000000 x48 + 4000000 x49 + 1000000 x50
+ 3000000 x51 + 3000000 x52 + 1000000 x53 + 1000000 x54 + 1000000 x55
+ 2000000 x56 + 2000000 x57 + 1000000 x58 + 3000000 x59 + 2000000 x60
+ 2400 x61 + 2000 x62 + 2600 x63 + 1800 x64 + 2600 x65 + 2250 x66
+ 1950 x67 + 2700 x68 + 1850 x69 + 2050 x70 + 2150 x71 + 2200 x72
+ 2050 x73 + 2000 x74 + 2100 x75 + 1950 x76 + 1950 x77 + 1900 x78
+ 2150 x79 + 2300 x80

Subject To

c1: x61 = 1
c2: x62 = 1
c3: x63 = 1
c4: x64 = 1
c5: x65 = 1
c6: x66 = 1
c7: x67 = 1
c8: x68 = 1
c9: x69 = 1
c10: x70 = 1
c11: x71 = 1
c12: x72 = 1
c13: x73 = 1
c14: x74 = 1
c15: x75 = 1
c16: x76 = 1
c17: x77 = 1

c18: x78	= 1	c57: x37	= 1
c19: x79	= 1	c58: x38	= 1
c20: x80	= 1	c59: x39	= 1
c21: x1	= 1	c60: x40	= 1
c22: x2	= 1	c61: x41	= 1
c23: x3	= 1	c62: x42	= 1
c24: x4	= 1	c63: x43	= 1
c25: x5	= 1	c64: x44	= 1
c26: x6	= 1	c65: x45	= 1
c27: x7	= 1	c66: x46	= 1
c28: x8	= 1	c67: x47	= 1
c29: x9	= 1	c68: x48	= 1
c30: x10	= 1	c69: x49	= 1
c31: x11	= 1	c70: x50	= 1
c32: x12	= 1	c71: x51	= 1
c33: x13	= 1	c72: x52	= 1
c34: x14	= 1	c73: x53	= 1
c35: x15	= 1	c74: x54	= 1
c36: x16	= 1	c75: x55	= 1
c37: x17	= 1	c76: x56	= 1
c38: x18	= 1	c77: x57	= 1
c39: x19	= 1	c78: x58	= 1
c40: x20	= 1	c79: x59	= 1
c41: x21	= 1	c80: x60	= 1
c42: x22	= 1		
c43: x23	= 1		
c44: x24	= 1		
c45: x25	= 1		
c46: x26	= 1		
c47: x27	= 1		
c48: x28	= 1		
c49: x29	= 1		
c50: x30	= 1		
c51: x31	= 1		
c52: x32	= 1		
c53: x33	= 1		
c54: x34	= 1		
c55: x35	= 1		
c56: x36	= 1		

End

Εικόνα 4.4: Εξισώσεις του RMP

Λύνοντας το παραπάνω RMP βρίσκουμε τιμή αντικειμενικής συνάρτησης $Z=146044550$ μονάδες. Στην συνέχεια θα πρέπει να λύσουμε και το column generation sub-problem. Όπως έχουμε ήδη αναφέρει το column generation sub-problem δεν θα το λύσουμε ως ξεχωριστό πρόβλημα, αλλά θα χρησιμοποιήσουμε ένα δίκτυο και θα εφαρμόσουμε έναν αλγόριθμο συντομότερης διαδρομής για μη κυκλικά δίκτυα. Αυτό στο λογισμικό επιτυγχάνεται με την δημιουργία συναρτήσεων.

Αρχικά δημιουργούμε μια συνάρτηση για την αρχικοποίηση των τιμών των κόμβων και στην συνέχεια με μια άλλη συνάρτηση κάνουμε τον έλεγχο για τον συνδυασμό των κόμβων και την δημιουργία του προγράμματος. Αφού δημιουργηθούν όλα τα δυνατά προγράμματα για κάθε πιλότο, επιλέγονται 3 από αυτά και εισάγονται στο master problem, το οποίο και λύνεται ξανά. Λύνοντας ξανά το master problem προκύπτουν νέες δυικές τιμές, οι οποίες αντικαθιστούν τις προηγούμενες. Έχοντας τις νέες δυικές τιμές, λύνουμε ξανά το δίκτυο, ώστε να βρούμε τα νέα προγράμματα για κάθε πιλότο. Η διαδικασία αυτή συνεχίζεται έως ότου να μην υπάρχει αρνητικό reduce cost. Όταν κανένα από αυτά τα προγράμματα δε θα έχει αρνητικό reduce cost, τότε θα ολοκληρωθεί το column generation sub-problem.

Στην εικόνα 4.5 φαίνεται το παραπάνω πρόβλημα όταν ολοκληρώνεται η πρώτη επανάληψη του column generation sub-problem, όπου για κάθε πιλότο έχουν εισαχθεί 3 μεταβλητές, οι οποίες αντιστοιχούν σε 3 προγράμματα. Όταν η μεταβλητή του προγράμματος ενός πιλότου εμφανίζεται σε έναν περιορισμό, ο οποίος αφορά μια διαδρομή, τότε η διαδρομή αυτή καλύπτεται από το πρόγραμμα αυτό. Ωστόσο όλοι οι περιορισμοί ισούνται με την μονάδα, πράγμα το οποίο σημαίνει ότι για τους περιορισμούς που αφορούν τους πιλότους μόνο ένα πρόγραμμα θα επιλεγεί για κάθε έναν από αυτούς. Αντίστοιχα και για τους περιορισμούς των διαδρομών σημαίνει ότι κάθε διαδρομή είτε θα μείνει ακάλυπτη, είτε θα ανήκει σε αποκλειστικά και μόνο σε ένα πρόγραμμα. Επειδή ο αριθμός των μεταβλητών, οι οποίες εισέρχονται στο πρόβλημα μετά από κάθε επανάληψη είναι μεγάλος, θα παρουσιάσουμε το πρόβλημα με την μορφή εξισώσεων μόνο για την πρώτη επανάληψη, καθώς και στο τέλος του column generation sub-problem, δηλαδή πριν το βήμα του branching.

Minimize

obj: 3000000 x1 + 3000000 x2 + 1000000 x3 + 1000000 x4 + 4000000 x5
+ 4000000 x6 + 1000000 x7 + 2000000 x8 + 4000000 x9 + 2000000 x10
+ 3000000 x11 + 4000000 x12 + 1000000 x13 + 2000000 x14 + 4000000 x15
+ 3000000 x16 + 1000000 x17 + 3000000 x18 + 2000000 x19 + 3000000 x20
+ 2000000 x21 + 2000000 x22 + 2000000 x23 + 2000000 x24 + 1000000 x25
+ 1000000 x26 + 2000000 x27 + 4000000 x28 + 2000000 x29 + 1000000 x30
+ 1000000 x31 + 2000000 x32 + 1000000 x33 + 4000000 x34 + 1000000 x35
+ 3000000 x36 + 2000000 x37 + 2000000 x38 + 2000000 x39 + 2000000 x40
+ 4000000 x41 + 2000000 x42 + 1000000 x43 + 4000000 x44 + 4000000 x45
+ 3000000 x46 + 3000000 x47 + 3000000 x48 + 4000000 x49 + 1000000 x50
+ 3000000 x51 + 3000000 x52 + 1000000 x53 + 1000000 x54 + 1000000 x55
+ 2000000 x56 + 2000000 x57 + 1000000 x58 + 3000000 x59 + 2000000 x60
+ 2400 x61 + 2000 x62 + 2600 x63 + 1800 x64 + 2600 x65 + 2250 x66
+ 1950 x67 + 2700 x68 + 1850 x69 + 2050 x70 + 2150 x71 + 2200 x72
+ 2050 x73 + 2000 x74 + 2100 x75 + 1950 x76 + 1950 x77 + 1900 x78
+ 2150 x79 + 2300 x80 + 1850 x81 + 1450 x82 + 11800 x83 + 1550 x84
+ 1700 x85 + 2200 x86 + 2150 x87 + 2050 x88 + 5800 x89 + 10000 x90
+ 1500 x91 + 1250 x92 + 6100 x93 + 2300 x94 + 1650 x95 + 5150 x96
+ 1950 x97 + 1300 x98 + 1500 x99 + 1650 x100 + 5950 x101 + 1750 x102
+ 1950 x103 + 7700 x104 + 1550 x105 + 4250 x106 + 1550 x107 + 1150 x108
+ 1600 x109 + 4150 x110 + 7450 x111 + 1850 x112 + 1600 x113 + 7750 x114
+ 1650 x115 + 1250 x116 + 6850 x117 + 1750 x118 + 1500 x119 + 5200 x120
+ 1250 x121 + 1350 x122 + 4600 x123 + 1350 x124 + 1450 x125 + 3750 x126
+ 1400 x127 + 1000 x128 + 5300 x129 + 1650 x130 + 1000 x131 + 1450 x132
+ 1350 x133 + 7500 x134 + 1700 x135 + 1200 x136 + 3700 x137 + 5950 x138
+ 2000 x139 + 1350 x140

Subject To

c1: x61 + x81 + x82 + x83 = 1
c2: x62 + x84 + x85 + x86 = 1
c3: x63 + x87 + x88 + x89 = 1
c4: x64 + x90 + x91 + x92 = 1
c5: x65 + x93 + x94 + x95 = 1
c6: x66 + x96 + x97 + x98 = 1
c7: x67 + x99 + x100 + x101 = 1
c8: x68 + x102 + x103 + x104 = 1
c9: x69 + x105 + x106 + x107 = 1
c10: x70 + x108 + x109 + x110 = 1
c11: x71 + x111 + x112 + x113 = 1

c12: $x_{72} + x_{114} + x_{115} + x_{116} = 1$
 c13: $x_{73} + x_{117} + x_{118} + x_{119} = 1$
 c14: $x_{74} + x_{120} + x_{121} + x_{122} = 1$
 c15: $x_{75} + x_{123} + x_{124} + x_{125} = 1$
 c16: $x_{76} + x_{126} + x_{127} + x_{128} = 1$
 c17: $x_{77} + x_{129} + x_{130} + x_{131} = 1$
 c18: $x_{78} + x_{132} + x_{133} + x_{134} = 1$
 c19: $x_{79} + x_{135} + x_{136} + x_{137} = 1$
 c20: $x_{80} + x_{138} + x_{139} + x_{140} = 1$
 c21: $x_1 + x_{84} + x_{87} + x_{89} + x_{90} + x_{93} + x_{99} + x_{101} + x_{111} + x_{114} + x_{129} + x_{132} + x_{135} = 1$
 c22: $x_2 + x_{85} + x_{91} + x_{94} + x_{97} + x_{100} + x_{105} + x_{112} + x_{118} + x_{130} + x_{139} = 1$
 c23: $x_3 + x_{81} + x_{83} + x_{88} + x_{92} + x_{113} + x_{115} + x_{117} + x_{119} + x_{126} + x_{127} + x_{133} + x_{134} = 1$
 c24: $x_4 + x_{82} + x_{83} + x_{95} + x_{96} + x_{98} + x_{102} + x_{104} + x_{116} + x_{117} + x_{126} + x_{128} + x_{131} + x_{136} + x_{137} + x_{138} + x_{140} = 1$
 c25: $x_5 + x_{86} + x_{96} + x_{106} + x_{117} + x_{120} + x_{134} + x_{137} + x_{138} = 1$
 c26: $x_6 + x_{123} + x_{126} = 1$
 c27: $x_7 + x_{83} + x_{90} + x_{93} + x_{103} + x_{104} + x_{111} + x_{114} + x_{121} + x_{124} = 1$
 c28: $x_8 + x_{83} + x_{89} + x_{90} + x_{93} + x_{101} + x_{104} + x_{108} + x_{110} + x_{129} = 1$
 c29: $x_9 + x_{86} + x_{89} + x_{93} + x_{104} + x_{109} + x_{110} + x_{126} + x_{129} + x_{138} = 1$
 c30: $x_{10} + x_{83} + x_{96} + x_{101} + x_{114} + x_{120} + x_{122} + x_{123} + x_{125} + x_{134} + x_{137} = 1$
 c31: $x_{11} + x_{106} + x_{107} + x_{114} + x_{117} + x_{137} = 1$
 c32: $x_{12} = 1$
 c33: $x_{13} + x_{83} + x_{90} + x_{101} + x_{111} + x_{120} + x_{134} = 1$
 c34: $x_{14} + x_{83} + x_{90} + x_{96} + x_{101} + x_{120} + x_{134} = 1$
 c35: $x_{15} + x_{90} + x_{101} + x_{123} + x_{134} = 1$
 c36: $x_{16} + x_{96} = 1$
 c37: $x_{17} + x_{83} + x_{93} + x_{106} + x_{111} + x_{114} + x_{117} + x_{129} = 1$
 c38: $x_{18} + x_{89} + x_{111} + x_{117} = 1$
 c39: $x_{19} + x_{93} + x_{138} = 1$
 c40: $x_{20} + x_{86} + x_{106} + x_{114} + x_{126} + x_{129} + x_{137} = 1$
 c41: $x_{21} + x_{83} + x_{104} + x_{110} + x_{120} = 1$
 c42: $x_{22} + x_{83} + x_{89} + x_{93} + x_{96} + x_{101} + x_{106} + x_{114} + x_{117} + x_{120} + x_{123} + x_{138} = 1$
 c43: $x_{23} + x_{86} + x_{90} + x_{104} + x_{110} + x_{111} + x_{126} + x_{134} = 1$
 c44: $x_{24} + x_{83} + x_{89} + x_{90} + x_{101} + x_{104} + x_{106} + x_{110} + x_{114} + x_{123} + x_{134} + x_{137} = 1$

c45: $x_{25} + x_{86} + x_{93} + x_{120} + x_{126} = 1$
c46: $x_{26} + x_{83} + x_{89} + x_{96} + x_{104} + x_{110} + x_{111} + x_{114} + x_{120} + x_{129} + x_{138} = 1$
c47: $x_{27} + x_{83} + x_{86} + x_{93} + x_{96} + x_{101} + x_{106} + x_{126} + x_{137} = 1$
c48: $x_{28} + x_{89} + x_{111} + x_{114} + x_{117} = 1$
c49: $x_{29} + x_{104} + x_{123} + x_{129} + x_{134} + x_{138} = 1$
c50: $x_{30} + x_{83} + x_{86} + x_{90} + x_{93} + x_{101} + x_{104} + x_{106} + x_{123} + x_{134} + x_{137} = 1$
c51: $x_{31} + x_{83} + x_{90} + x_{93} + x_{96} + x_{104} + x_{120} + x_{129} + x_{134} + x_{138} = 1$
c52: $x_{32} + x_{83} + x_{93} + x_{96} + x_{104} + x_{106} + x_{110} + x_{123} + x_{129} + x_{134} + x_{137} = 1$
c53: $x_{33} + x_{90} + x_{120} = 1$
c54: $x_{34} + x_{90} + x_{93} + x_{96} + x_{101} + x_{123} + x_{126} + x_{129} + x_{137} = 1$
c55: $x_{35} + x_{83} + x_{86} + x_{110} + x_{111} + x_{114} + x_{117} = 1$
c56: $x_{36} + x_{104} + x_{117} = 1$
c57: $x_{37} + x_{106} + x_{114} + x_{120} + x_{138} = 1$
c58: $x_{38} + x_{83} + x_{86} + x_{111} + x_{134} = 1$
c59: $x_{39} + x_{101} + x_{110} + x_{111} + x_{117} + x_{134} = 1$
c60: $x_{40} + x_{83} + x_{86} + x_{89} + x_{90} + x_{96} + x_{104} + x_{106} + x_{120} + x_{126} = 1$
c61: $x_{41} + x_{89} + x_{111} + x_{123} + x_{137} = 1$
c62: $x_{42} + x_{83} + x_{86} + x_{90} + x_{101} + x_{106} + x_{110} + x_{114} + x_{117} + x_{134} + x_{138} = 1$
c63: $x_{43} + x_{83} + x_{90} + x_{93} + x_{96} + x_{104} + x_{106} + x_{114} + x_{120} + x_{138} = 1$
c64: $x_{44} + x_{86} + x_{90} + x_{93} + x_{96} + x_{101} + x_{106} + x_{114} + x_{117} + x_{120} + x_{129} + x_{138} = 1$
c65: $x_{45} + x_{110} + x_{134} = 1$
c66: $x_{46} = 1$
c67: $x_{47} = 1$
c68: $x_{48} = 1$
c69: $x_{49} + x_{89} + x_{104} + x_{111} + x_{123} + x_{126} = 1$
c70: $x_{50} + x_{83} + x_{86} + x_{90} + x_{101} + x_{110} + x_{114} + x_{117} + x_{134} + x_{138} = 1$
c71: $x_{51} + x_{106} + x_{134} + x_{137} = 1$
c72: $x_{52} + x_{86} + x_{93} + x_{96} + x_{101} + x_{104} + x_{110} + x_{111} + x_{120} + x_{123} + x_{126} + x_{129} + x_{138} = 1$
c73: $x_{53} + x_{83} + x_{89} = 1$
c74: $x_{54} + x_{83} + x_{89} + x_{90} + x_{114} + x_{117} = 1$
c75: $x_{55} + x_{83} + x_{90} + x_{114} = 1$
c76: $x_{56} + x_{83} + x_{89} + x_{106} + x_{110} + x_{120} + x_{123} + x_{129} + x_{134} = 1$
c77: $x_{57} + x_{138} = 1$

c78: $x_{58} + x_{104} + x_{111} + x_{117} + x_{126} = 1$
c79: $x_{59} + x_{86} + x_{90} + x_{96} + x_{101} + x_{111} + x_{114} + x_{117} + x_{126} = 1$
c80: $x_{60} + x_{83} + x_{89} + x_{90} + x_{110} + x_{111} + x_{114} + x_{138} = 1$
End

Εικόνα 4.5 Εξισώσεις RMP μετά την I'' επανάληψη του column generation.

Στην εικόνα 4.5 παρατηρούμε ότι ο αριθμός των μεταβλητών από 80 που ήταν πριν, τώρα έγινε 140. Αυτό είναι λογικό καθώς είπαμε ότι για κάθε πιλότο (έχουμε 20 πιλότους) εισέρχονται 3 μεταβλητές στο πρόβλημα. Οι πρώτοι 20 περιορισμοί έχουν 4 μεταβλητές. Η μια είναι του κενού προγράμματος και οι άλλες τρεις λόγω της πρώτης επανάληψης του column generation sub-problem. Οι υπόλοιποι περιορισμοί περιέχουν τα προγράμματα, στα οποία καλύπτεται η αντίστοιχη διαδρομή, καθώς και τη μεταβλητή της ακάλυπτης διαδρομής. Λύνοντας αυτό το πρόβλημα μας δίνει τιμή αντικειμενικής συνάρτησης $Z1= 46600024.194$ μονάδες, τιμή αρκετά μικρότερη με την τιμή του αρχικού προβλήματος ($Z=146044550$ μονάδες). Επομένως παρατηρούμε ότι με την εισαγωγή επιπλέον μεταβλητών στο πρόβλημα η λύση βελτιώθηκε. Παρακάτω παραθέτουμε τις λύσεις της αντικειμενικής συνάρτησης για κάθε επανάληψη του column generation sub-problem. Οι λύσεις αυτές είναι:

$$Z2= 34344147.059$$

$$Z3= 24092717.225$$

$$Z4= 16457512.607$$

$$Z5= 10757309.235$$

$$Z6= 6976891.027$$

$$Z7= 2875670.643$$

$$Z8= 195846.108$$

$$Z9= 33661.211$$

$$Z10= 30255.511$$

$$Z11= 25961.506$$

$$Z12= 22130.092$$

$$Z13= 19528.113$$

$$Z14= 17380.023$$

$$Z15= 16273.889$$

$$Z16= 13699.701$$

$$Z17= 12316.376$$

$$Z18= 10861.281$$

$$Z19= 10365.964$$

$$Z20= 9789.229$$

$$Z21= 9476.191$$

$$Z22= 9399.100$$

$$Z23= 9399.999$$

Βλέποντας τις παραπάνω τιμές, παρατηρούμε ότι η τιμή της αντικειμενικής συνάρτησης μειώνεται συνεχώς, καθώς προσθέτουμε μεταβλητές στο master problem. Η μείωση αυτή είναι πολύ μεγάλη, αφού από $Z=146044550$ μονάδες, κάνοντας 23 επαναλήψεις, μειώθηκε στις $Z23= 9399.999$ μονάδες. Ο αριθμός των μεταβλητών του προβλήματος αυξήθηκε στις 1460 μεταβλητές. Επειδή το πρόβλημα που προκύπτει είναι πολύ μεγάλο για να απεικονίσουμε όλες τις εξισώσεις, στο παράρτημα φαίνεται η αντικειμενική συνάρτηση του προβλήματος και οι 4 πρώτοι περιορισμοί, προκειμένου να γίνει αντιληπτό για τι τάξης μεγέθους εξισώσεις έπρεπε να λυθούν.

Με την επίλυση του MP βρήκαμε μια λύση $Z=9399.999$ μονάδες. Όμως αυτή δεν είναι η βέλτιστη λύση, διότι η τιμή των μεταβλητών απόφασης δεν είναι ακέραιοι αριθμοί. Γι'αυτό και θα περάσουμε στο επόμενο βήμα, στο βήμα του Branching, προκειμένου να βρούμε την βέλτιστη ακέραια λύση του προβλήματος. Κατά το Branching θα λύνουμε μόνο τα προβλήματα, στα οποία θέτεται η τιμή της μεταβλητής ίση με 1. Η παραπάνω λύση που βρήκαμε αποτελεί το κάτω όριο του δέντρου, δηλαδή δεν μπορούμε σε κανέναν κλάδο του δέντρου να βρούμε καλύτερη λύση από αυτή. Σε κάθε κλάδο θα θέτουμε την μεταβλητή με την μεγαλύτερη δεκαδική τιμή ίση με 1 και στην συνέχεια θα λύνεται το πρόβλημα, βρίσκοντας και πάλι νέα προγράμματα για κάθε ιπτάμενο, δηλαδή λύνοντας και πάλι το column generation sub-problem. Για κάθε μεταβλητή που θέτουμε ίση με 1, θα διαγράφονται

από το πρόβλημα όλες οι άλλες μεταβλητές που αντιστοιχούν στον περιορισμό του ιπταμένου, αλλά και στους περιορισμούς των διαδρομών που αυτή καλύπτει, εκτός από την μεταβλητή που δηλώνει εάν η διαδρομή έμεινε ακάλυπτη.. Προκειμένου όμως να μην ξαναδημιουργήσει το πρόβλημα τις ίδιες μεταβλητές, εισάγονται στον κώδικα κάποιοι περιορισμοί, οι οποίοι όμως αφορούν τον τρόπο εκτέλεσης του κώδικα και δεν επηρεάζουν τους περιορισμούς του προβλήματος. Έτσι επιτυγχάνεται σε κάθε κλάδο να έχουμε και ένα διαφορετικό πρόβλημα, αλλά με ίδιο αριθμό περιορισμών.

Η μεταβλητή με την μεγαλύτερη δεκαδική τιμή είναι η μεταβλητή 1073, με τιμή 0.75 και αντιστοιχεί στον 11 πιλότο. Έτσι στον πρώτο κλάδο του δέντρου λύνεται το πρόβλημα, όπου η μεταβλητή αυτή θέτεται ίση με 1 και διαγράφονται από το πρόβλημα όλες οι άλλες μεταβλητές που αντιστοιχούν στον ίδιο ιπτάμενο. Διαγράφοντας αυτές τις μεταβλητές και λύνοντας το πρόβλημα προσθέτοντας και άλλες, το index, δηλαδή ο αριθμός της μεταβλητής αλλάζει. Όμως, αλλάζει μόνο ο αριθμός της μεταβλητής και τίποτα άλλο. Οι περιορισμοί και ο πιλότος, στους οποίους αναφέρεται δεν επηρεάζονται καθόλου. Στην εικόνα 4.6 φαίνεται ότι η μεταβλητή 1073 πήρε την τιμή 1 και όλες οι άλλες μεταβλητές του περιορισμού διαγράφηκαν. Επίσης διαγράφηκαν και όλες οι άλλες μεταβλητές που υπήρχαν στους περιορισμούς που αυτό το πρόγραμμα κάλυψε, όπως φαίνεται στην εικόνα 4.7. Έτσι το νέο index της μεταβλητής 1073 είναι 927, δηλαδή η είναι η ίδια μεταβλητή με πριν αλλά με διαφορετικό όνομα.

$$\begin{aligned}
c10: & x70 + x105 + x106 + x107 + x156 + x157 + x158 + x203 + x204 + x246 + x247 \\
& + x290 + x291 + x342 + x343 + x344 + x395 + x396 + x397 + x446 + x447 \\
& + x448 + x501 + x502 + x503 + x554 + x555 + x556 + x604 + x605 + x606 \\
& + x657 + x658 + x659 + x713 + x714 + x766 + x767 + x818 + x819 + x820 \\
& + x875 + x876 + x877 + x925 + x926 + x977 + x978 + x979 + x1033 + x1034 \\
& + x1035 + x1090 + x1091 + x1092 + x1145 + x1146 + x1147 + x1195 + x1196 \\
& + x1197 + x1247 + x1248 + x1249 = 1
\end{aligned}$$

$$c11: x927 = 1$$

$$\begin{aligned}
c12: & x71 + x108 + x109 + x159 + x160 + x205 + x206 + x248 + x249 + x292 + x293 \\
& + x294 + x345 + x346 + x347 + x398 + x399 + x449 + x450 + x504 + x505 \\
& + x506 + x557 + x558 + x607 + x608 + x660 + x661 + x662 + x715 + x716 \\
& + x717 + x768 + x769 + x821 + x822 + x823 + x878 + x879 + x880 + x928 \\
& + x929 + x980 + x981 + x982 + x1036 + x1037 + x1038 + x1093 + x1094 \\
& + x1095 + x1148 + x1149 + x1150 + x1198 + x1199 + x1250 + x1251 + x1252 \\
& = 1
\end{aligned}$$

Εικόνα 4.6: Διαγραφή μεταβλητών από τον περιορισμό του πιλότου.

$$\begin{aligned}
c31: & x11 + x104 + x128 + x320 + x329 + x334 + x341 + x347 + x350 + x359 + x362 \\
& + x365 + x368 + x796 + x805 + x811 + x817 + x823 + x826 + x835 + x838 \\
& + x841 + x844 + x847 + x874 + x976 + x1067 + x1125 + x1150 + x1153 \\
& + x1161 + x1178 + x1194 + x1209 + x1212 + x1246 = 1
\end{aligned}$$

$$c32: x12 + x927 = 1$$

$$\begin{aligned}
c33: & x13 + x82 + x88 + x99 + x125 + x132 + x140 + x150 + x153 + x170 + x173 \\
& + x175 + x382 + x421 + x427 + x430 + x439 + x442 + x455 + x461 + x464 \\
& + x467 + x532 + x538 + x550 + x563 + x569 + x572 + x689 + x695 + x698 \\
& + x706 + x708 + x717 + x723 + x729 + x732 + x735 + x779 + x799 + x814 \\
& + x856 + x958 + x973 + x988 + x993 + x999 + x1014 + x1029 + x1222 + x1240 \\
& + x1241 = 1
\end{aligned}$$

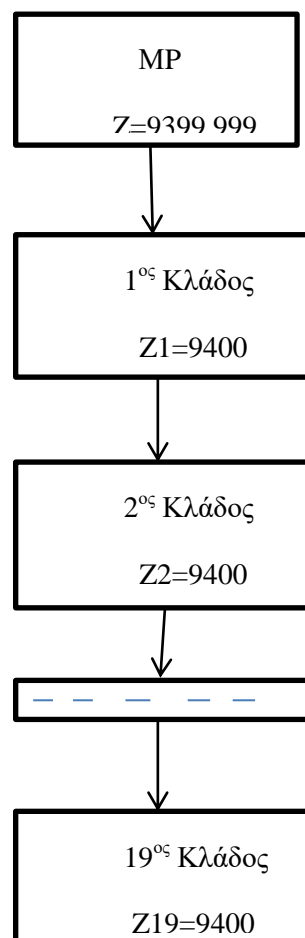
Εικόνα 4.7: Διαγραφή μεταβλητών από περιορισμό διαδρομής.

Λύνοντας το πρόβλημα αυτό μας δίνει τιμή αντικειμενικής συνάρτησης $Z=9400.000$ μονάδες. Στην συνέχεια λύνουμε το column generation sub-problem, όπως και πριν. Θα παρουσιάσουμε μόνο τις τιμές της αντικειμενικής συνάρτησης, οι οποίες προκύπτουν σε κάθε επανάληψη του column generation sub-problem. Οι τιμές αυτές είναι:

➤ $Z1=9400.000$

Παρατηρούμε ότι έκανε μόνο μια επανάληψη και το αποτέλεσμα της αντικειμενικής συνάρτησης ήταν το ίδιο και δεν είχαμε καλύτερη τιμή στην αντικειμενική συνάρτηση.

Για να περάσουμε στον δεύτερο κλάδο του δέντρου θα πρέπει να βρούμε στην λύση του προβλήματος του πρώτου κλάδου την μεταβλητή με την μεγαλύτερη δεκαδική τιμή. Έχοντας αυτή την μεταβλητή θα περάσουμε στον δεύτερο κλάδο, όπου θα η μεταβλητή θα πάρει την τιμή 1 και θα λύσουμε πάλι το πρόβλημα. Η διαδικασία αυτή συνεχίζεται έως ότου να μην υπάρχει άλλη δεκαδική τιμή στην λύση. Τότε θα έχουμε βρει την βέλτιστη ακέραιη λύση του προβλήματος. Το δέντρο διακλάδωσης του προβλήματος φαίνεται στο σχήμα 4.6.



Σχήμα 4.6: *Branching tree.*

Στην εικόνα 4.8 φαίνεται η τελική λύση του προβλήματος, όπως επιλέχθηκε από εμάς, να την εκτυπώνει το λογισμικό IBM ILOG CPLEX 12.5.1. Παρατηρούμε ότι ο κύριος στόχος του προβλήματος επιτεύχθηκε, δηλαδή όλες οι διαδρομές καλύφθηκαν. Ένα μέρος του κόστους οφείλεται στο ότι το πρόγραμμα για να βγάλει την βέλτιστη λύση έδωσε στον πιλότο 13 το κενό πρόγραμμα. Το υπόλοιπο οφείλεται στο γεγονός ότι για κάποιους πιλότους, οι συνολικές ώρες πτήσης τους είναι εκτός του διαστήματος που τους αντιστοιχεί.

```

the flight window of pilot 1 is LHR=48 UHR=56
the flight window of pilot 2 is LHR=40 UHR=46
the flight window of pilot 3 is LHR=52 UHR=62
the flight window of pilot 4 is LHR=36 UHR=43
the flight window of pilot 5 is LHR=52 UHR=60
the flight window of pilot 6 is LHR=45 UHR=55
the flight window of pilot 7 is LHR=39 UHR=46
the flight window of pilot 8 is LHR=54 UHR=63
the flight window of pilot 9 is LHR=37 UHR=46
the flight window of pilot 10 is LHR=41 UHR=50
the flight window of pilot 11 is LHR=43 UHR=50
the flight window of pilot 12 is LHR=44 UHR=51
the flight window of pilot 13 is LHR=41 UHR=48
the flight window of pilot 14 is LHR=40 UHR=49
the flight window of pilot 15 is LHR=42 UHR=48
the flight window of pilot 16 is LHR=39 UHR=46
the flight window of pilot 17 is LHR=39 UHR=49
the flight window of pilot 18 is LHR=38 UHR=47
the flight window of pilot 19 is LHR=43 UHR=50
the flight window of pilot 20 is LHR=46 UHR=53

The solution of the problem as a integer problem is objective_value=9400.000000

The uncovered routes are:
The pilot 13 has the routes: 0 with total flight time: 0 hours
The pilot 4 has the routes: 3 with total flight time: 11 hours
The pilot 12 has the routes: 4 with total flight time: 19 hours
The pilot 19 has the routes: 15 29 35 with total flight time: 41 hours
The pilot 2 has the routes: 6 14 30 38 49 57 with total flight time: 39 hours
The pilot 15 has the routes: 24 48 50 52 with total flight time: 41 hours
The pilot 16 has the routes: 7 16 59 with total flight time: 36 hours
The pilot 18 has the routes: 18 58 with total flight time: 34 hours
The pilot 11 has the routes: 12 28 37 with total flight time: 43 hours
The pilot 8 has the routes: 13 19 23 36 with total flight time: 53 hours
The pilot 20 has the routes: 26 31 45 with total flight time: 44 hours
The pilot 3 has the routes: 1 53 54 60 with total flight time: 52 hours
The pilot 5 has the routes: 17 41 55 with total flight time: 52 hours
The pilot 6 has the routes: 5 10 33 51 with total flight time: 44 hours
The pilot 7 has the routes: 34 47 with total flight time: 31 hours
The pilot 10 has the routes: 9 25 32 42 46 with total flight time: 37 hours
The pilot 1 has the routes: 21 with total flight time: 4 hours
The pilot 14 has the routes: 22 39 43 56 with total flight time: 37 hours
The pilot 17 has the routes: 2 8 with total flight time: 24 hours
The pilot 9 has the routes: 11 20 27 40 44 with total flight time: 29 hours

the solution was achieved after 19 times of branching

```

Εικόνα 4.8: Τελική λύση προβλήματος.

Παρόλα αυτά η βέλτιστη λύση βρέθηκε κάνοντας 19 φορές branching. Στην εικόνα 4.9 παρουσιάζουμε τις τελικές εξισώσεις, η λύση των οποίων οδήγησε στην παραπάνω λύση.

Minimize

obj: 3000000 x1 + 3000000 x2 + 1000000 x3 + 1000000 x4 + 4000000 x5
+ 4000000 x6 + 1000000 x7 + 2000000 x8 + 4000000 x9 + 2000000 x10
+ 3000000 x11 + 4000000 x12 + 1000000 x13 + 2000000 x14 + 4000000 x15
+ 3000000 x16 + 1000000 x17 + 3000000 x18 + 2000000 x19 + 3000000 x20
+ 2000000 x21 + 2000000 x22 + 2000000 x23 + 2000000 x24 + 1000000 x25
+ 1000000 x26 + 2000000 x27 + 4000000 x28 + 2000000 x29 + 1000000 x30
+ 1000000 x31 + 2000000 x32 + 1000000 x33 + 4000000 x34 + 1000000 x35
+ 3000000 x36 + 2000000 x37 + 2000000 x38 + 2000000 x39 + 2000000 x40
+ 4000000 x41 + 2000000 x42 + 1000000 x43 + 4000000 x44 + 4000000 x45
+ 3000000 x46 + 3000000 x47 + 3000000 x48 + 4000000 x49 + 1000000 x50
+ 3000000 x51 + 3000000 x52 + 1000000 x53 + 1000000 x54 + 1000000 x55
+ 2000000 x56 + 2000000 x57 + 1000000 x58 + 3000000 x59 + 2000000 x60
+ 1850 x61 + 2050 x62 + 1250 x63 + 1550 x64 + 1250 x65 + 100 x66 + 50 x67
+ 50 x68 + 1550 x69 + 150 x70 + 200 x71 + 50 x73 + 1550 x74 + 100 x75
+ 50 x78 + 1550 x79 + 1550 x80 + 400 x81 + 1550 x82 + 1550 x83 + 200 x84
+ 2200 x85 + 1550 x86 + 1550 x87 + 1550 x88 + 1550 x89 + 1550 x90
+ 1550 x91 + 1550 x92 + 1550 x93 + 1550 x94 + 1550 x95 + 1550 x96
+ 150 x97 + 1550 x98 + 1550 x99 + 750 x100 + 1550 x101 + 950 x102
+ 1550 x103 + 850 x104 + 1550 x105 + 1550 x106 + 1550 x107 + 1600 x108
+ 500 x109 + 1550 x110 + 700 x111 + 1300 x112 + 1550 x113 + 1600 x114
+ 400 x115

Subject To

c1: x85 = 1
c2: x67 = 1
c3: x76 = 1
c4: x63 = 1
c5: x77 = 1
c6: x78 = 1
c7: x81 = 1
c8: x73 = 1
c9: x61 + x64 + x69 + x74 + x79 + x80 + x82 + x83 + x86 + x87 + x88 + x89
+ x90 + x91 + x92 + x93 + x94 + x95 + x96 + x98 + x99 + x101 + x102
+ x103 + x104 + x105 + x106 + x107 + x108 + x109 + x110 + x111 + x112
+ x113 + x114 + x115 = 1
c10: x84 = 1
c11: x72 = 1
c12: x65 = 1
c13: x62 = 1

c14: $x_{97} = 1$
 c15: $x_{68} = 1$
 c16: $x_{70} = 1$
 c17: $x_{100} = 1$
 c18: $x_{71} = 1$
 c19: $x_{66} = 1$
 c20: $x_{75} = 1$
 c21: $x_1 + x_{76} = 1$
 c22: $x_2 + x_{100} = 1$
 c23: $x_3 + x_{63} = 1$
 c24: $x_4 + x_{65} = 1$
 c25: $x_5 + x_{78} = 1$
 c26: $x_6 + x_{67} = 1$
 c27: $x_7 + x_{70} = 1$
 c28: $x_8 + x_{100} = 1$
 c29: $x_9 + x_{84} = 1$
 c30: $x_{10} + x_{78} = 1$
 c31: $x_{11} + x_{64} + x_{69} + x_{74} + x_{79} + x_{80} + x_{82} + x_{83} + x_{86} + x_{87} + x_{88} + x_{89}$
 $+ x_{90} + x_{91} + x_{92} + x_{93} + x_{94} + x_{95} + x_{96} + x_{98} + x_{99} + x_{101} + x_{103}$
 $+ x_{104} + x_{105} + x_{106} + x_{107} + x_{109} + x_{110} + x_{112} + x_{113} + x_{115} = 1$
 c32: $x_{12} + x_{72} = 1$
 c33: $x_{13} + x_{73} = 1$
 c34: $x_{14} + x_{67} = 1$
 c35: $x_{15} + x_{66} = 1$
 c36: $x_{16} + x_{70} = 1$
 c37: $x_{17} + x_{77} = 1$
 c38: $x_{18} + x_{71} = 1$
 c39: $x_{19} + x_{73} = 1$
 c40: $x_{20} + x_{104} + x_{108} + x_{109} + x_{111} + x_{112} + x_{114} + x_{115} = 1$
 c41: $x_{21} + x_{85} = 1$
 c42: $x_{22} + x_{97} = 1$
 c43: $x_{23} + x_{73} = 1$
 c44: $x_{24} + x_{68} = 1$
 c45: $x_{25} + x_{84} = 1$
 c46: $x_{26} + x_{75} = 1$
 c47: $x_{27} + x_{102} + x_{104} + x_{111} + x_{115} = 1$
 c48: $x_{28} + x_{72} = 1$
 c49: $x_{29} + x_{66} = 1$
 c50: $x_{30} + x_{67} = 1$

```

c51: x31 + x75 = 1
c52: x32 + x84 = 1
c53: x33 + x78 = 1
c54: x34 + x81 = 1
c55: x35 + x66 = 1
c56: x36 + x73 = 1
c57: x37 + x72 = 1
c58: x38 + x67 = 1
c59: x39 + x97 = 1
c60: x40 + x102 + x104 + x109 + x111 + x115 = 1
c61: x41 + x77 = 1
c62: x42 + x84 = 1
c63: x43 + x97 = 1
c64: x44 + x102 + x109 + x111 + x115 = 1
c65: x45 + x75 = 1
c66: x46 + x84 = 1
c67: x47 + x81 = 1
c68: x48 + x68 = 1
c69: x49 + x67 = 1
c70: x50 + x68 = 1
c71: x51 + x78 = 1
c72: x52 + x68 = 1
c73: x53 + x76 = 1
c74: x54 + x76 = 1
c75: x55 + x77 = 1
c76: x56 + x97 = 1
c77: x57 + x67 = 1
c78: x58 + x71 = 1
c79: x59 + x70 = 1
c80: x60 + x76 = 1
End

```

Εικόνα 4.9: Τελικές εξισώσεις του προβλήματος.

Το παραπάνω πρόβλημα ήταν ένα πρόβλημα 20 πιλότων, 60 διαδρομών. Παρατηρήσαμε ότι έφτασε να έχει 1460 μεταβλητές, επιλέγοντας 3 path. Τα πραγματικά προβλήματα μεγάλων αεροπορικών εταιριών είναι πολύ μεγαλύτερα από αυτό. Ωστόσο, δεν θα μπορούσε εδώ να αναλυθεί βήμα προς βήμα ένα τόσο μεγάλο πρόβλημα διότι, θα απαιτούσε πολλές επαναλήψεις και θα είχε μεγάλο αριθμό μεταβλητών. Σκοπός του παραδείγματος είναι να γίνει κατανοητό από τον αναγνώστη η μεθοδολογία επίλυσης του προβλήματος, οπότε το μικρότερο πρόβλημα που διαλέξαμε είναι περισσότερο αντιπροσωπευτικό για τον σκοπό αυτό.

ΚΕΦΑΛΑΙΟ 5

ΥΠΟΛΟΓΙΣΤΙΚΗ ΕΜΠΕΙΡΙΑ

5.1 Εισαγωγή

Η ανάλυση του αλγοριθμικού τρόπου επίλυσης του προβλήματος παρουσιάστηκε εκτενώς στο κεφάλαιο 4. Κατά την επίλυση του προβλήματος λάβαμε υπόψη και τους διάφορους παράγοντες, οι οποίοι μπορούν να επηρεάσουν τη λύση. Τα βήματα που ακολουθήσαμε για τον αλγοριθμικό τρόπο επίλυσης είναι η αρχικοποίηση του RMP, η επίλυση του master LP relaxation, ο καθορισμός των μεταβλητών που εισάγονται ανά επανάληψη, η επίλυση του column generation sub-problem και τέλος η εφαρμογή του βήματος Branching. Η επίλυση του master LP relaxation problem έγινε με τη βοήθεια του λογισμικού IBM ILOG CPLEX 12.5.1, ενώ η επίλυση του column generation sub-problem έγινε με αλγόριθμο συντομότερης διαδρομής για μη κυκλικά δίκτυα.

Προκειμένου να ελεγχθεί η ορθότητα και η αποτελεσματικότητα του αλγορίθμου που αναπτύχθηκε, θα παρουσιάσουμε στη συνέχεια μία σειρά από παραδείγματα, τα οποία θα συμβάλλουν στο να εξεταστεί ο αλγόριθμος στις διαφορετικές περιπτώσεις μεγέθους δεδομένων, αλλά και παραμέτρων και να γίνει η σύγκριση και η ανάλυση των αποτελεσμάτων τους. Τα παραδείγματα αυτά διαφέρουν ως προς το μέγεθος του πληρώματος (N), το μέγεθος των διαδρομών/routes (R), καθώς και ως προς τη παράμετρο paths. Σε κάθε παράδειγμα εισάγονται διαφορετικές τιμές στα παραπάνω και εξετάζουμε τα αποτελέσματα. Το κόστος για κάθε διαδρομή που μένει ακάλυπτη το θέτουμε στις 1000000 μονάδες, ενώ 50 μονάδες θα είναι το κόστος για τις ώρες εκτός του διαστήματος των ωρών πτήσης, τις οποίες πρέπει να καλύψει κάθε ιπτάμενος.

Τα παραδείγματα που εξετάζονται είναι δεν αντιστοιχούν σε κάποια πραγματική αεροπορική εταιρεία, καθώς όπως αναφέραμε και στην παράγραφο 4.7, οι τιμές του διαστήματος των ωρών πτήσης κάθε πιλότου, της διάρκειας κάθε διαδρομής, της ημέρας αναχώρησης και προσγείωσης κάθε διαδρομής, καθώς και των συμβατών διαδρομών για κάθε πιλότο αλλά και των διαδρομών με τις οποίες μπορούν να συνδυαστούν, προκύπτουν τυχαία μέσω μιας γεννήτριας τυχαίων αριθμών. Παρόλα αυτά, λόγω του ότι τα παραδείγματα που θα ακολουθήσουν καλύπτουν ένα μεγάλο εύρος ως προς το μέγεθος των προβλημάτων, ο αλγόριθμος που αναπτύχθηκε μπορεί κάλλιστα να χρησιμοποιηθεί και για ένα πραγματικό πρόβλημα μιας αεροπορικής εταιρείας.

Όλα τα παραδείγματα επιλύθηκαν μέσω του λογισμικού IBM ILOG CPLEX 12.5.1. Πραγματοποιήθηκαν σε ένα Intel Core i3 @ 2.20 GHz με μνήμη συστήματος στα 4 GB. Τα τεχνικά χαρακτηριστικά αυτού του συστήματος είναι σημαντικά κατώτερα από τα χαρακτηριστικά των συστημάτων που χρησιμοποιούνται σε ρεαλιστικό περιβάλλον. Σε ένα ρεαλιστικό περιβάλλον, όπου χρησιμοποιούνται μηχανήματα με καλύτερα χαρακτηριστικά αναμένεται πως ο προτεινόμενος αλγόριθμος επίλυσης θα έχει σημαντικά καλύτερη επίδοση.

Στον παρακάτω πίνακα παρουσιάζονται συγκεντρωτικά τα αποτελέσματα από όλα τα παραδείγματα που θα ακολουθήσουν. Για κάθε μια περίπτωση μεγέθους επιλύθηκαν είκοσι (20) διαφορετικά προβλήματα. Η επίλυση πολλών διαφορετικών προβλημάτων ίδιου μεγέθους είναι αναγκαία, προκειμένου να αποδειχθεί η αποτελεσματικότητα και η σταθερότητα των αποτελεσμάτων του αλγορίθμου. Γενικότερα, στην βιβλιογραφία προτείνεται να γίνονται από 20 έως 30 επαναλήψεις για πιο αξιόπιστα αποτελέσματα. Στον παρακάτω πίνακα οι τιμές που παρουσιάζονται είναι οι μέσες τιμές που προέκυψαν από την επίλυση των είκοσι διαφορετικών προβλημάτων σε κάθε περίπτωση.

		Time	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered Routes	# Branchi ng
N=100 R=400	paths=1	4:57	17223	17426	203	0	98
N=150 R=900	paths =1	2:55:14	68439	68690	250	0	148
N=90 R=750	paths =1	1:02:05	147632	147636	4	0	89
N=110 R=750	paths =1	1:06:43	105667	105744	77	0	109
N=90 R=550	paths=1	20:57	47610	147633	100022	0	89
	paths=2	37:23	50555	50780	224	0	89
	paths=3	42:48	53465	53650	184	0	89
	paths=5	1:19:35	45438	46160	721	0	89
	paths=10	2:17:42	46779	46890	165	0	89

Πίνακας 5.1: Συγκεντρωτικός πίνακας αποτελεσμάτων

5.2 Εφαρμογές του αλγορίθμου σε προβλήματα διαφορετικού μεγέθους

5.2.1 Εφαρμογή 1^η

Δεδομένα που εισάχθηκαν στον αλγόριθμο: N=100, R=400, paths=1

Στην πρώτη εφαρμογή δώσαμε στη μεταβλητή N, που δείχνει τον αριθμό των πιλότων, την τιμή 100, ενώ στη μεταβλητή R η οποία είναι ο αριθμός των διαδρομών που πρέπει να καλυφθούν, την τιμή 400 και θέσαμε paths=1. Τα αποτελέσματα που λάβαμε παρουσιάζονται παρακάτω στον πίνακα 5.2.

	time	Λύση MP	Τελική βέλτιστη λύση	gap	Branching
min	0:04:13	7750	7750	0	97
max	0:06:05	31600	31750	800	99
average	0:04:57	17223	17426	203	98

Πίνακας 5.2: Αποτελέσματα 1^{ης} εφαρμογής.

Το πρόβλημα επιλύθηκε κατά μέσο όρο σε 4 λεπτά και 57 δευτερόλεπτα με μέγιστο χρόνο τα 6 λεπτά και 5 δευτερόλεπτα και ελάχιστο τα 4 λεπτά και 13 δευτερόλεπτα. Η μέση τιμή του MP είναι 17223, με ελάχιστη τιμή 7750 και μέγιστη 31600. Η τιμή της αντικειμενικής συνάρτησης του τελικού προβλήματος μετά το branching, κυμαίνεται από 7750 μέχρι 31750, με μέση τιμή 17426. Η τιμή της διαφοράς της τιμής του MP από την τιμή του τελικού προβλήματος ονομάζεται gap και ήταν αρκετές φορές 0 με μέγιστη τιμή το 800 και μέση 203. Τέλος, ο αριθμός των branching κυμάνθηκε από 97 μέχρι 99. Στο συγκεκριμένο παράδειγμα δεν

υπήρξε καμία περίπτωση ακάλυπτης διαδρομής. Βέβαια, ο χρόνος επίλυσης του προβλήματος ήταν πολύ μικρός και για αυτό το λόγο στην επόμενη εφαρμογή χρησιμοποιήθηκαν μεγαλύτερες τιμές, τόσο ως προς τον αριθμό των πιλότων όσο και ως προς τον αριθμό των διαδρομών.

5.2.2 Εφαρμογή 2^η

Δεδομένα που εισάχθηκαν στον αλγόριθμο: N=150, R=900, paths=1

Στην εφαρμογή του αλγορίθμου στην περίπτωση αυτή, αυξήσαμε τον αριθμό των πιλότων σε N=150, από N=100 που ήταν πριν και τον αριθμό των διαδρομών σε R=900, από R=400 που ήταν πριν, ενώ την τιμή του paths την κρατήσαμε σταθερή. Η εφαρμογή αυτή είναι πιο πολύπλοκη σε σχέση με την προηγούμενη και αναμένεται να αυξηθεί ο χρόνος επίλυσης.

	time	Λύση MP	Τελική βέλτιστη λύση	gap	branching
min	2:33:54	63365	64150	-178	147
max	3:28:24	73729	73550	785	149
average	2:55:14	68439	68690	250	148

Πίνακας 5.3: Αποτελέσματα 2^{ης} εφαρμογής.

Το πρόβλημα επιλύθηκε σε 2ω 55λ και 14δ κατά μέσο όρο με μέγιστο χρόνο επίλυσης τις 3ω 28λ και 24δ και ελάχιστο τις 2ω 33λ και 54δ. Η μέγιστη τιμή του MP είναι 73729, η ελάχιστη 63365 και η μέση τιμή 68439. Η βέλτιστη τελική τιμή είναι κατά μέσο όρο 68690, με μέγιστη τιμή 73550 και ελάχιστη 64150. Το gap κυμαίνεται από -178 έως 785 με μέσο όρο το 250, ενώ ο αριθμός των κόμβων του

δέντρου, δηλαδή ο αριθμός των branching κυμαίνεται από 147 μέχρι 149. Και σε αυτή την περίπτωση δεν υπήρξε καμία διαδρομή που να μην καλύφθηκε. Πρώτη φορά συναντάμε στα προβλήματα που επιλύθηκαν έως τώρα αρνητικό gap, δηλαδή η τιμή του τελικού προβλήματος είναι μικρότερη από αυτή του MP. Τυπικά αυτό δε μπορεί να συμβεί, καθώς η λύση του MP μας δίνει τη μικρότερη λύση του προβλήματος, καθώς έχουν βρεθεί όλοι οι συνδυασμοί που μειώνουν την τιμή της αντικειμενικής και επίσης έχει επιτραπεί στις μεταβλητές να πάρουν και δεκαδικές τιμές. Η εμφάνιση αρνητικού gap έχει να κάνει κυρίως με τον αριθμό των path. Όπως έχουμε εξηγήσει και στην παράγραφο 4.4, κάποιες φορές μια μεταβλητή που εισάγεται στο πρόβλημα και θεωρείται ως η καλύτερη, αλλά το τελικό πρόβλημα ίσως μια άλλη μεταβλητή, η οποία δεν θεωρούνταν αρχικά η καλύτερη να το βελτιώνει περισσότερο. Για την εξάλειψη αυτού του φαινομένου θα δοθούν μεγαλύτερες τιμές στη μεταβλητή paths, όμως η συγκεκριμένη εφαρμογή είναι ήδη αρκετά μεγάλη και ο χρόνος επίλυσής των προβλημάτων της θα αυξηθεί αρκετά. Τέλος, αυτό που παρατηρούμε είναι ότι με την προσθήκη επιπλέον διαδρομών και περισσότερων πιλότων τα προβλήματα γίνονται πιο περίπλοκα και αυξάνεται σημαντικά ο χρόνος επίλυσής τους.

5.2.3 Εφαρμογή 3^η

Δεδομένα που εισάχθηκαν στον αλγόριθμο: $N=90$, $R=750$, $paths=1$

Στην εφαρμογή αυτή μειώσαμε τον αριθμό των πιλότων, αλλά κρατήσαμε υψηλό τον αριθμό των διαδρομών και την τιμή $paths=1$. Έπειτα από είκοσι φορές που επιλύσαμε το πρόβλημα μέσω του αλγορίθμου που αναπτύξαμε και χρησιμοποιώντας γεννήτρια τυχαίων τιμών πήραμε τα αποτελέσματα που παρουσιάζονται στον πίνακα 5.4.

	time	Λύση MP	Τελική βέλτιστη λύση	gap	branching
min	0:42:35	140300	140300	0	89
max	1:31:33	162926	162950	24	89
average	1:02:05	147632	147636	3	89

Πίνακας 5.4: Αποτελέσματα 3^{ης} εφαρμογής.

Τα προβλήματα της συγκεκριμένης εφαρμογής επιλύθηκαν κατά μέσο όρο σε 1ω 2λ και 5δ με μέγιστο χρόνο τη 1ω 31λ και 33δ και ελάχιστο χρόνο τα 42λ και 35δ. Η μέγιστη τιμή του MP είναι 162926 και η ελάχιστη 140300 με μέση τιμή 147632. Αντίστοιχα Η τελική βέλτιστη λύση κυμαίνεται από 140300 έως 162950 με μέσο όρο 147636. Οι τιμές του master problem και της τελικής βέλτιστης λύσης είναι στις περισσότερες περιπτώσεις ίδιες, καθώς η τιμή του gap είναι πολύ μικρή. Παρατηρούμε ότι οι τιμές της αντικειμενικής συνάρτησης της τελικής βέλτιστης λύσης έχουν αυξηθεί κατά πολύ σε σχέση με τα προβλήματα της προηγούμενης εφαρμογής και αυτό συμβαίνει γιατί άλλαξε η αναλογία μεταξύ πιλότων και διαδρομών. Έτσι, υπάρχει ένα μεγάλο κόστος των πρόσθετων διαδρομών που θα πρέπει να εκτελέσουν οι πιλότοι. Από τη στιγμή που σε καμία επανάληψη δεν εμφανίστηκε αρνητικό gap, δε θα εξετάσουμε σε αυτή την εφαρμογή τη λύση των προβλημάτων με διαφορετικές τιμές στη μεταβλητή paths. Τέλος, οι τιμές του branching είναι και στα είκοσι προβλήματα, τα οποία επιλύθηκαν, 89 και σε όλα καλύφθηκαν όλες οι διαδρομές.

5.2.4 Εφαρμογή 4^η

Δεδομένα που εισάχθηκαν στον αλγόριθμο: N=110, R=750, paths=1

Στα προβλήματα αυτής της εφαρμογής αυξήσαμε τον αριθμό των πιλότων σε 110, ενώ κρατήσαμε τον αριθμό των διαδρομών στις 750, ώστε να δούμε την επίδραση στο χρόνο που έχει η αύξηση των πιλότων. Έπειτα από την επίλυση είκοσι διαφορετικών προβλημάτων με αριθμούς της τυχαίας γεννήτριας, πήραμε τα αποτελέσματα που φαίνονται παρακάτω στον πίνακα 5.5.

	time	Λύση MP	Τελική βέλτιστη λύση	gap	branching
Min	0:53:45	98966	99050	0	109
Max	1:24:55	112816	112850	341	109
Average	1:06:43	105667	105744	77	109

Πίνακας 5.5: Αποτελέσματα 4^{ης} εφαρμογής.

Ο μέγιστος χρόνος επίλυσης είναι 1ω 24λ και 55δ, ενώ ο ελάχιστος 53λ και 45δ. Ο μέσος χρόνος επίλυσης των προβλημάτων είναι 1ω 6λ και 43δ. Όπως ήταν αναμενόμενο, ο χρόνος επίλυσης αυξήθηκε αλλά όχι σημαντικά. Οι τιμές του MP κυμαίνονται από 98966 έως 112816 με μέση τιμή 105667. Η μέση τελική βέλτιστη τιμή είναι 105744, με 99050 και 112850 η ελάχιστη και η μέγιστη τιμή αντίστοιχα. Παρατηρείται, επίσης, οι τελικές βέλτιστες τιμές είναι μικρότερες σε σχέση με την προηγούμενη εφαρμογή, καθώς αυξάνοντας τον αριθμό των πιλότων επιμερίστηκαν καλύτερα οι διαδρομές μέσα στο διάστημα ωρών πτήσεών τους. Η μέση τιμή του gap είναι 77, ενώ η ελάχιστη είναι 0 και η μέγιστη διαφορά 341. Η τιμή του branching είναι 109 σε όλα τα προβλήματα. Όλες οι διαδρομές καλύφθηκαν σε όλα τα προβλήματα, κάτι το οποίο φαίνεται και από τις τιμές των λύσεων του πίνακα 5.5.

5.2.5 Εφαρμογή 5^η

Δεδομένα που εισάχθηκαν στο πρόβλημα: N=90, R=550, paths=1.

Εδώ ο αριθμός των πιλότων είναι ξανά N=90, μειώσαμε όμως τον αριθμό των διαδρομών σε R=550 και κρατήσαμε σταθερό και το paths=1. Τα αποτελέσματα που προκύπτουν φαίνονται στον πίνακα 5.6.

	time	Λύση MP	Τελική βέλτιστη λύση	gap	branching
min	0:14:21	38919	38850	-257	88
max	0:30:38	58015	2041100	1999425	89
average	0:20:57	47610	147633	100022	89

Πίνακας 5.6: Αποτελέσματα 5^{ης} εφαρμογής.

Κρατώντας σταθερό τον αριθμό των πιλότων και μειώνοντας τον αριθμό των διαδρομών, βλέπουμε πως τα προβλήματα επιλύονται σε πολύ μικρότερο χρόνο. Ο μέγιστος χρόνος επίλυσης είναι τα 30λ και 38δ ενώ ο ελάχιστος χρόνος είναι τα 14λ και 21δ. Ο μέσος χρόνος επίλυσης των προβλημάτων είναι τα 20λ και 57δ. Οι τιμές του MP κυμαίνονται από 38919 έως 58015, με μέση τιμή 47610.

Στη συγκεκριμένη εφαρμογή, κατά την επίλυση του όγδοου προβλήματος στο βήμα του branching μία διαδρομή δεν καλύφθηκε και αυτό έχει ως συνέπεια μεγάλη αύξηση στην τιμή της αντικειμενικής συνάρτησης, δηλαδή στο κόστος της υποθετικής αεροπορικής εταιρίας. Για αυτή τη μία διαδρομή που δεν καλύφθηκε το κόστος αυξήθηκε κατά 2000000 μονάδες και για αυτό το λόγο η τελική βέλτιστη τιμή της αντικειμενικής στο πρόβλημα αυτό είναι 2041100. Η ελάχιστη τελική βέλτιστη τιμή της αντικειμενικής συνάρτησης είναι 38919, ενώ ο μέσος όρος, ο οποίος έχει επηρεαστεί από την παραπάνω ακραία τιμή της αντικειμενικής στην περίπτωση που δεν καλύφθηκαν όλες οι διαδρομές, είναι στα 147633. Τέλος, η

ελάχιστη τιμή του gap είναι -257 και η μέγιστη 1999425, η οποία οφείλεται και αυτή στη διαδρομή που δε καλύφθηκε, ενώ το branching είναι 88 ή 89 σε όλες τις επαναλήψεις με μέσο όρο το 89. Εφόσον στο παράδειγμά μας εμφανίστηκε αρνητικό gap σε περισσότερες από μία περιπτώσεις θεωρήθηκε ως το ιδανικό, προκειμένου να εξεταστεί ως προς την αλλαγή της παραμέτρου paths.

Οι λύσεις των 20 προβλημάτων αυτής της εφαρμογής παρουσιάζονται συγκεντρωτικά στο Παράρτημα 2.

5.2.6 Εφαρμογή 6^η

Δεδομένα που εισάχθηκαν στο πρόβλημα: N=90, R=550.

Στη συγκεκριμένη εφαρμογή κρατήσαμε τις ίδιες τιμές για τον αριθμό των ιπταμένων και των διαδρομών, N=90 και R=550 αντίστοιχα, και μεταβάλλαμε την τιμή της παραμέτρου paths. Οι τιμές που δόθηκαν είναι 2,3,5 και 10.

5.2.6.1 Τιμή παραμέτρου paths=2

	time	Λύση MP	Τελική βέλτιστη λύση	gap	branching
Min	0:28:22	42640	42700	-86	89
Max	0:49:27	55286	55450	814	89
Average	0:37:23	50555	50780	224	89

Πίνακας 5.7: Αποτελέσματα 6^{ης} εφαρμογής για paths=2.

Στην περίπτωση αυτή, όπου paths=2, ο μέσος χρόνος επίλυσης των προβλημάτων ήταν 37λ και 23δ με μέγιστο χρόνο τα 49λ και 27δ και ελάχιστο τα

28λ και 22δ. Η μέση τιμή του MP είναι 50555, με 42640 και 55286 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η τελική βέλτιστη τιμή της αντικειμενικής συνάρτησης κυμάνθηκε από 42700 έως 55450, με μέσο όρο 50780. Η ελάχιστη τιμή gap είναι -86, η μέγιστη 814 και ο μέσος όρος 224. Η τιμή του branching είναι σε όλα τα προβλήματα 89 και όλες οι πτήσεις καλύφθηκαν. Η αύξηση της παραμέτρου paths, αύξησε το χρόνο επίλυσης του προβλήματος σχεδόν στο διπλάσιο, ενώ και πάλι παρατηρήθηκε αρνητική διάφορα μεταξύ του master problem και της τελικής βέλτιστης λύσης.

5.2.6.2 Τιμή παραμέτρου paths=3

	time	Λύση MP	Τελική βέλτιστη λύση	gap	branching
Min	0:39:43	43423	43600	22	89
Max	0:51:46	65444	65650	372	89
Average	0:42:48	53465	53650	184	89

Πίνακας 5.8: Αποτελέσματα 6^{ης} εφαρμογής για paths=3.

Στην περίπτωση όπου paths=3, τα προβλήματα επιλύονται κατά μέσο όρο σε 42λ και 48δ. Ο μέγιστος χρόνος επίλυσης είναι 51λ και 46δ, ενώ ο ελάχιστος 39λ και 43δ. Οι τιμές του MP κυμάνθηκαν από 43423 έως 65444, με μέσο όρο 53465, ενώ η μέγιστη και η ελάχιστη τιμή της τελικής βέλτιστης λύσης είναι 43600 και 65650 αντίστοιχα. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 53650. Το gap είναι κατά μέσο όρο 184 με ελάχιστο το 22 και μέγιστο το 372, ενώ ο αριθμός των branching είναι και πάλι σε όλες τις επαναλήψεις ίσος με 89. Αντίθετα με την προηγούμενη περίπτωση, εδώ δεν έχουμε αρνητικές τιμές του gap. Η εξασφάλιση όμως της σωστής λύσης έχει ως αντίτιμο την αύξηση του χρόνου επίλυσής του, με τη

διαφορά βέβαια να μη διαφέρει πολύ από την προηγούμενη περίπτωση, καθώς έχει αυξηθεί κατά 5λεπτά.

5.2.6.3 Τιμή παραμέτρου $paths=5$

	time	ΛύσηMP	Τελική βέλτιστη λύση	gap	branching
Min	1:08:46	41480	42500	540	89
Max	1:34:35	54793	55550	1020	89
Average	1:19:35	45438	46160	721	89

Πίνακας 5.9: Αποτελέσματα 6^{ης} εφαρμογής για $paths=5$.

Θέτοντας την τιμή $paths=5$, ο μέσος χρόνος επίλυσης των προβλημάτων ανέρχεται στη 1ω 19 και 35δ. Ο μέγιστος και ο ελάχιστος χρόνος επίλυσης είναι η 1ω 34λ και 35δ και η 1ω 8λ και 46δ αντίστοιχα. Η μέγιστη τιμή του MP είναι 54793 και η ελάχιστη 41480, ενώ ο μέσος όρος είναι 41480. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 46160 με ελάχιστη τιμή 42500 και μέγιστη τιμή 55550. Οι τιμές του gap κυμάνθηκαν από 540 μέχρι 1020 με μέσο όρο 721, ενώ η τιμή του branching ήταν 89. Τέλος, σε όλα τα προβλήματα της περίπτωσης αυτής καλύφθηκαν όλες οι διαδρομές.

5.2.6.4 Τιμή παραμέτρου $paths=10$

	time	master	αντικειμενική	gap	branching
Min	1:41:36	44153	44000	50	89
Max	2:39:16	54218	54450	247	89
Average	2:17:42	46779	46890	165	89

Πίνακας 5.10: Αποτελέσματα 6^{ης} εφαρμογής για $paths=10$.

Στη συνέχεια, ορίσαμε την τιμή του $paths=10$. Ο μέγιστος χρόνος επίλυσης ήταν 2ω 39λ και 16δ, ενώ ο ελάχιστος 1ω 41λ και 36δ, με μέσο χρόνο επίλυσης τις 2ω 39λ και 16δ. Ο χρόνος επίλυσης των προβλημάτων είναι σημαντικά αυξημένος σε σχέση με τις προηγούμενες περιπτώσεις. Η τιμή του MP ήταν κατά μέσο όρο 46779, με μέγιστη τιμή 54218 και ελάχιστη τιμή 44153. Αντίστοιχα, οι τιμές της τελικής βέλτιστης λύσης ήταν κατά μέσο όρο 46890 με μέγιστο 54450 και ελάχιστο 44000. Το gap κυμάνθηκε από 50 έως 247, με μέσο όρο την τιμή 165. Ο αριθμός των branching και σε αυτή την περίπτωση είναι 89, ενώ καλύφθηκαν και πάλι όλες οι διαδρομές. Με εξαίρεση το χρόνο επίλυσης του παραδείγματος, βλέπουμε ότι τα αποτελέσματα του παραδείγματος είναι παρόμοια με τις προηγούμενες περιπτώσεις, οπότε θέτοντας μία τέτοια μεγάλη τιμή στη μεταβλητή $paths$ μπορεί απλά να κάνει πιο αργή την επίλυση του προβλήματος, χωρίς να οδηγήσει σε καλύτερη λύση. Επομένως γίνεται εύκολα αντιληπτό ότι δεν είναι η καλύτερη επιλογή να πάρει η παράμετρος $paths$ τόσο μεγάλη τιμή.

ΚΕΦΑΛΑΙΟ 6

ΣΥΝΟΨΗ, ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

6.1 Σύνοψη

Αντικειμενικός σκοπός της παρούσας διπλωματικής εργασίας ήταν η ανάπτυξη ενός αλγορίθμου ανάθεσης, ο οποίος αποσκοπούσε στην κάλυψη ενός συνόλου διαδρομών από ένα συγκεκριμένο αριθμό πιλότων με το ελάχιστο δυνατό κόστος, λαμβάνοντας ταυτόχρονα υπόψη κάποια ποιοτικά κριτήρια.

Αρχικά, στο πρώτο και δεύτερο κεφάλαιο εστιάσαμε στην περιγραφή του προβλήματος ανάθεσης προσωπικού από τις αεροπορικές εταιρίες. Επιπλέον, έγινε μία βιβλιογραφική αναφορά στις διάφορες μορφές που συναντάται το πρόβλημα αυτό καθώς και στις λύσεις του..

Στο τρίτο κεφάλαιο διατυπώθηκε το πρόβλημα το οποίο μελετήθηκε και αναπτύχθηκε το μαθηματικό μοντέλο το οποίο το περιγράφει. Στη συνέχεια στο επόμενο κεφάλαιο παρουσιάστηκε η αλγοριθμική προσέγγιση του προβλήματος και αναλύθηκε, βήμα προς βήμα, η μεθοδολογία επίλυσης με την βοήθεια ενός μικρού παραδείγματος..

Το πέμπτο κεφάλαιο αποτελεί και το πειραματικό μέρος της εργασίας. Σε αυτό έγινε η εφαρμογή του αλγορίθμου σε περιπτώσεις διαφορετικού μεγέθους προβλημάτων τόσο ως προς τον αριθμό των πιλότων και των διαδρομών που καλούνται να καλύψουν, όσο και ως προς τον αριθμό της παραμέτρου paths.

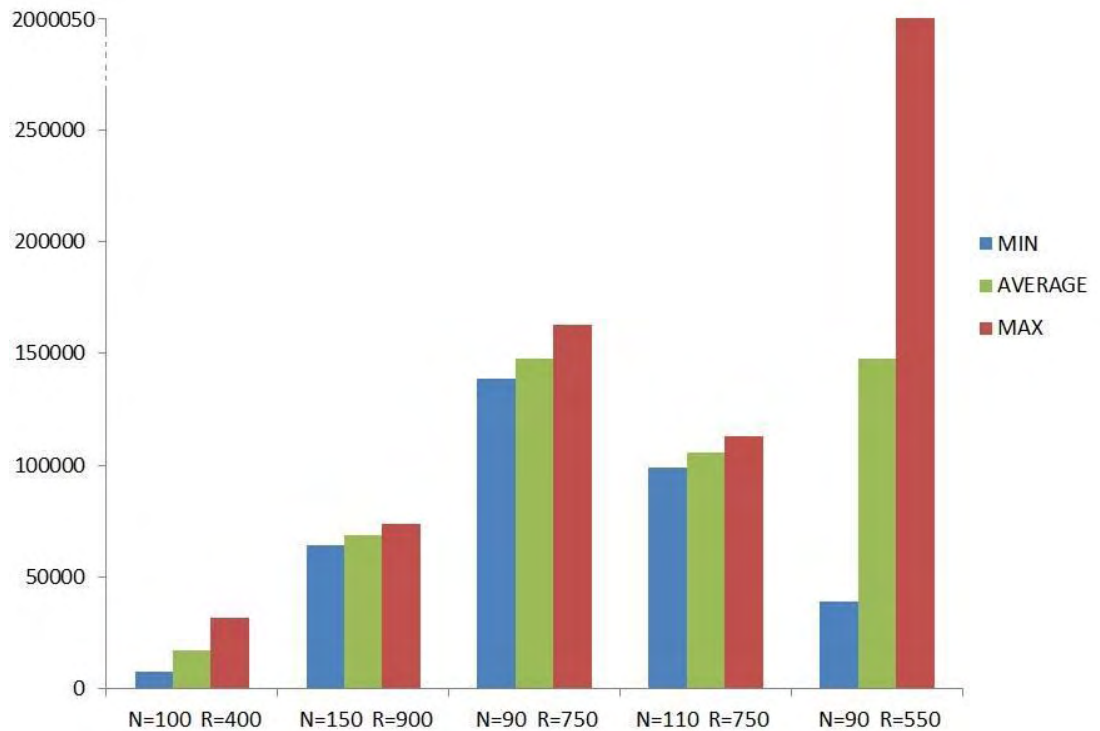
Τέλος, στο παρόν κεφάλαιο έγινε η συλλογή και η ανάλυση των αποτελεσμάτων που προέκυψαν στο κεφάλαιο 5, τα οποία μας οδήγησαν στα συμπεράσματα που

παρουσιάζονται παρακάτω, καθώς και σε προτάσεις για μελλοντική μελέτη του προβλήματος.

6.2 Συμπεράσματα

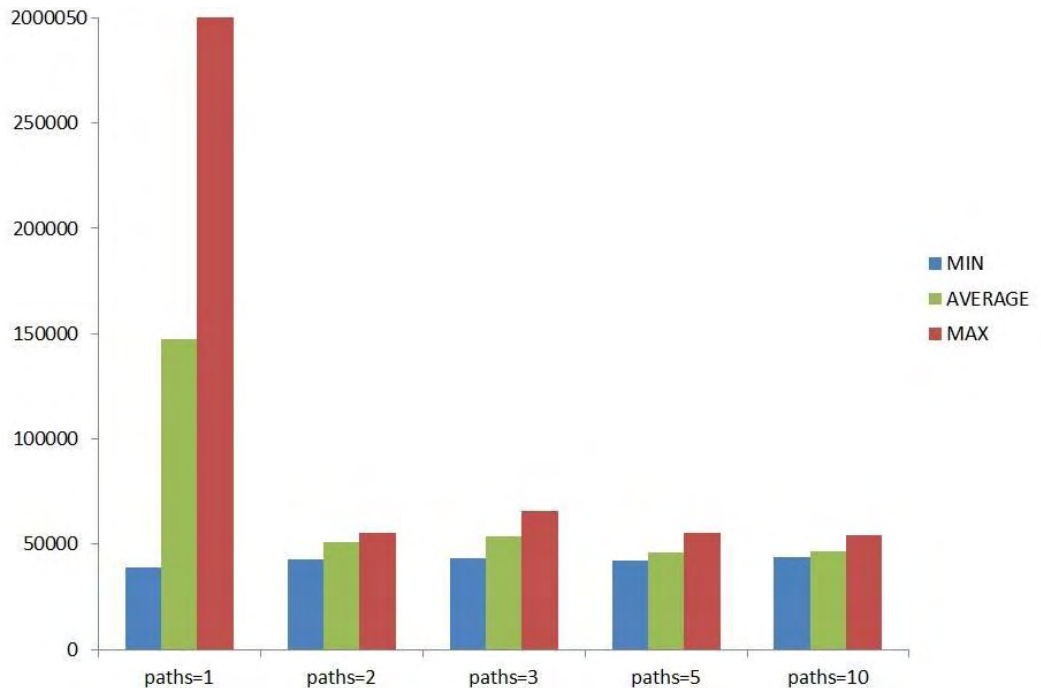
Έπειτα από την επιτυχή ολοκλήρωση των εφαρμογών του κεφαλαίου 5, παρατηρήσαμε, όπως ήταν και αναμενόμενο, διαφοροποίηση των αποτελεσμάτων, τόσο ως προς τον χρόνο επίλυσης των προβλημάτων, όσο και ως προς την τιμή της τελικής βέλτιστης λύσης.

Η διαφοροποίηση ως προς την τιμή της τελικής βέλτιστης λύσης, οφείλεται κατά κύριο λόγο στις διαφορετικές τιμές που δόθηκαν σε κάθε μια από τις εφαρμογές, στον αριθμό των πιλότων και στον αριθμό των διαδρομών. Παρατηρήσαμε όμως ότι οι τιμές πολλές φορές ήταν διαφορετικές ακόμα και για το ίδιο μέγεθος προβλημάτων. Αυτό συμβαίνει λόγω του ότι οι περισσότερες τιμές των περιορισμών παράγονται μέσω μιας γεννήτριας τυχαίων αριθμών. Παρόλα αυτά, οι περιπτώσεις των μεγάλων αποκλίσεων είναι πολύ λίγες σε σχέση με τον αριθμό των προβλημάτων που λύθηκαν στις εφαρμογές και αυτό φαίνεται από το μέσο όρο που κυμαίνεται πιο κοντά στις ελάχιστες τιμές των λύσεων. Αυτό φαίνεται και στο σχήμα 6.1.



Σχήμα 6.1: Ελάχιστη, μέση και μέγιστη τιμή της τελικής βέλτιστης λύσης των εφαρμογών, όπου $paths=1$.

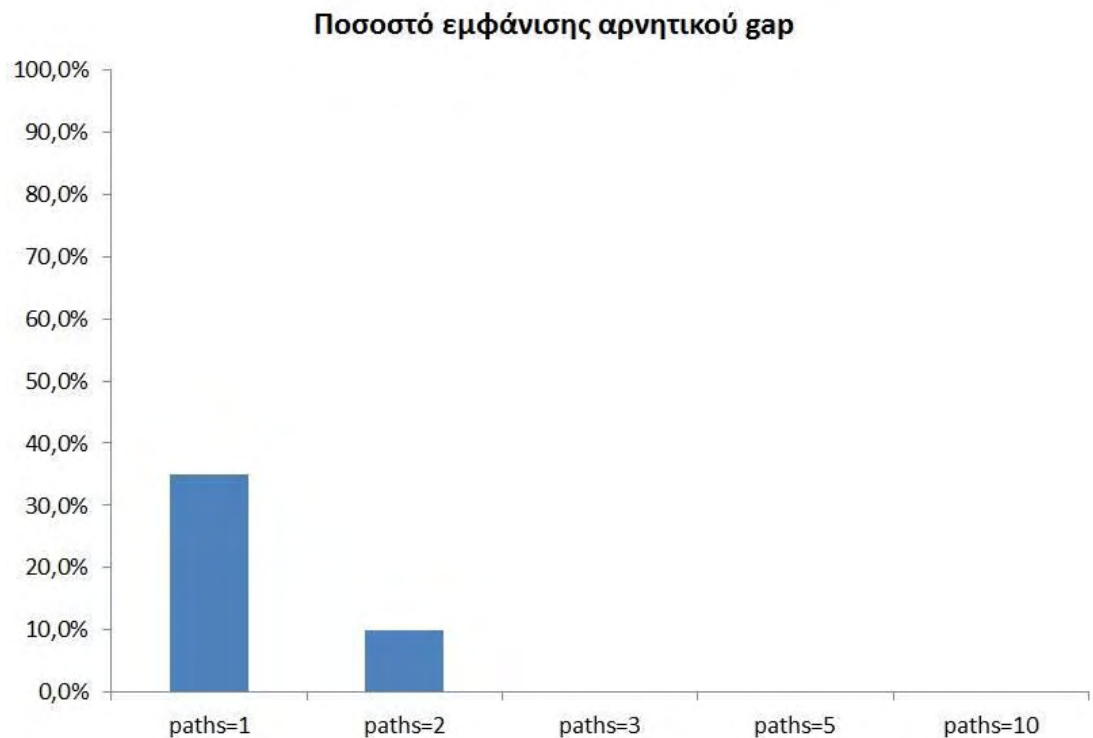
Στο σχήμα 6.1 παρουσιάζονται οι μεταβολές στην τελική βέλτιστη λύση, με την λεπτομέρεια ότι όλες οι εφαρμογές αφορούν προβλήματα που η παράμετρος $paths$ έχει την τιμή 1. Στο σχήμα 6.2 παρουσιάζουμε τις διαφοροποιήσεις ως προς την τιμή της τελικής βέλτιστης λύσης για τις διάφορες περιπτώσεις της εφαρμογή 6 ($N=90$, $R=450$), στις οποίες αλλάζει ο αριθμός της παραμέτρου $paths$, ενώ μένει ίδιος ο αριθμός των πιλότων και των διαδρομών.



Σχήμα 6.2: Ελάχιστη, μέση και μέγιστη τιμή της τελικής βέλτιστης λύσης για την εφαρμογή 6, για διάφορες τιμές της παραμέτρου *paths*.

Παρατηρούμε στο σχήμα 6.2 ότι μόνο για την περίπτωση που έχουμε ένα *paths* έχουμε ακραία τιμή. Σε όλες τις υπόλοιπες περιπτώσεις οι λύσεις και των είκοσι προβλημάτων σε κάθε περίπτωση ήταν πολύ κοντά μεταξύ τους. Αυτό μας δείχνει ότι η ακραία τιμή δεν οφείλεται στον αλγόριθμο που δημιουργήσαμε, αλλά στις τιμές που δημιουργούνται τυχαία από την γεννήτρια, οδηγώντας τον αλγόριθμο στο να επιλέξει να αφήσει μια διαδρομή ακάλυπτη προκειμένου να βρει την βέλτιστη λύση. Ωστόσο, αυτό έγινε μόνο σε ένα πρόβλημα από τα 180 προβλήματα που επιλύθηκαν συνολικά.

Επιπλέον, όπως είχαμε αναφέρει και στο κεφάλαιο 5, η εφαρμογή 5 θεωρήθηκε ως ιδανική για την μελέτη της επίδρασης της αλλαγής της τιμής της παραμέτρου *paths* στον αριθμό των προβλημάτων που παρουσιάζουν αρνητικό gap. Η επίδραση αυτή φαίνεται στο σχήμα 6.3.

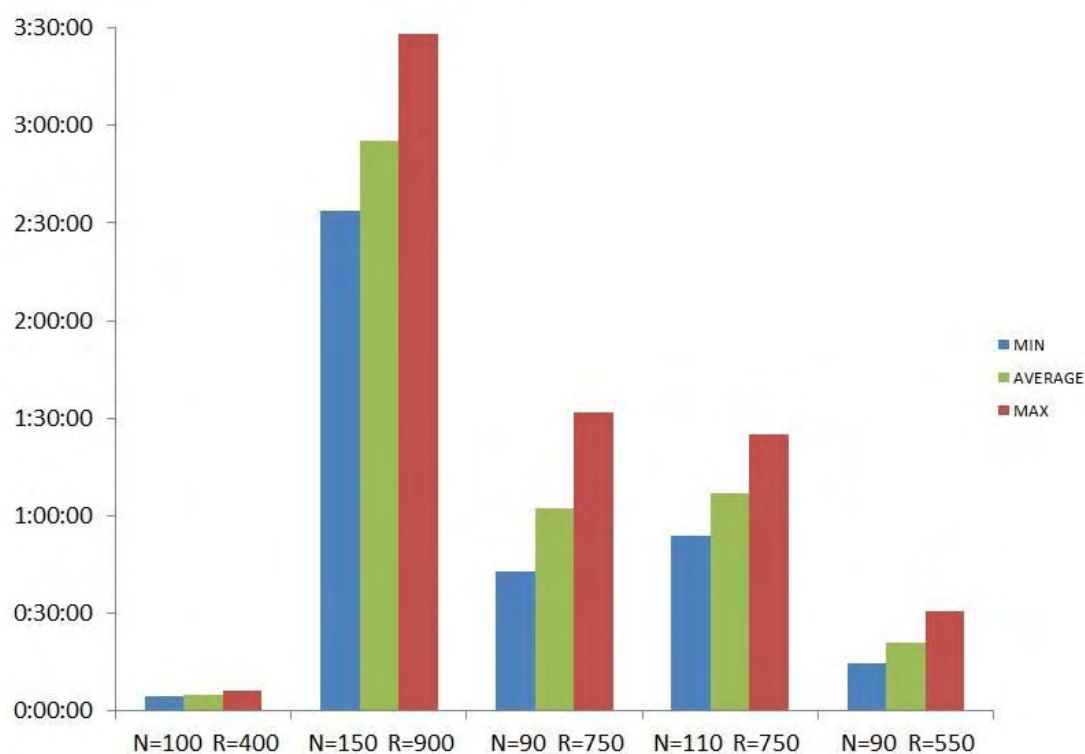


Σχήμα 6.3: Ποσοστό εμφάνισης αρνητικού gap για τις διάφορες τιμές της παραμέτρου paths.

Παρατηρούμε ότι όταν είχαμε paths=1 το 35% των προβλημάτων που λύθηκαν, δηλαδή 7 από τα 20 προβλήματα, παρουσίασαν αρνητικό gap, πράγμα το οποίο υποδηλώνει ότι στα προβλήματα αυτά ο αλγόριθμος δεν βρήκε όλα τα εφικτά προγράμματα των πιλότων, τα οποία θα μπορούσαν να βελτιώσουν την λύση συνολικά. Αυτό εξακολούθησε να γίνεται και όταν η παράμετρο paths πήρε την τιμή 2, δηλαδή σε κάθε επανάληψη του column generation, ο αλγόριθμος εύρισκε και εισήγαγε στο πρόβλημα τα δυο καλύτερα προγράμματα για κάθε πιλότο. Ωστόσο σε αυτήν την περίπτωση το ποσοστό είναι 10%, το οποίο είναι χαρακτηριστικά μικρότερο, καθώς μόλις 2 από τα 20 προβλήματα εμφάνισαν αρνητικό gap. Για τις τιμές 3, 5, 10 το ποσοστό ήταν 0%, δηλαδή κανένα πρόβλημα δεν εμφάνισε αρνητικό gap. Επομένως, γίνεται αντιληπτό ότι η επιλογή της τιμής της παραμέτρου paths πρέπει να γίνει πολύ προσεκτικά για κάθε πρόβλημα και μπορεί να διαφέρει για διαφορετικού μεγέθους προβλήματα. Όσο μεγαλύτερο είναι το πρόβλημα, τόσο μεγαλύτερη τιμή χρειάζεται, συνήθως, προκειμένου να βρεθεί η τελική βέλτιστη

λύση σωστά. Όμως, η αύξηση της τιμής της παραμέτρου *paths* αυξάνει τον χρόνο επίλυσης του προβλήματος. Πολλές φορές η επιλογή μεγαλύτερης τιμής αυξάνει τον χρόνο επίλυσης χωρίς να βελτιώνει ουσιαστικά την τελική λύση του προβλήματος. Επομένως, καταλαβαίνουμε ότι η επιλογή αυτής της τιμής δεν είναι απλή διαδικασία, καθώς αλληλεπιδρά με αρκετούς παράγοντες. Παρακάτω θα γίνει η ανάλυση της επίδρασης του μεγέθους του προβλήματος στον χρόνο επίλυσής του, σύμφωνα με τα αποτελέσματα των εφαρμογών του κεφαλαίου 5, καθώς και η επίδραση της τιμής της παραμέτρου *paths* στον χρόνο επίλυσης προβλημάτων ίδιου μεγέθους.

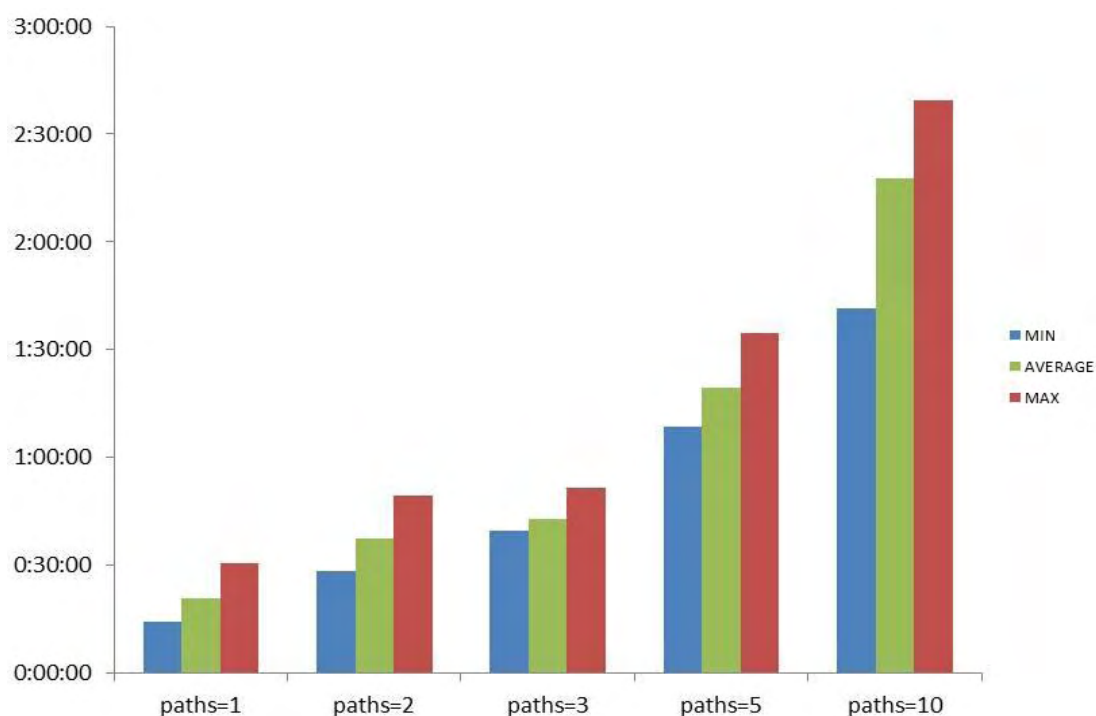
Στο σχήμα 6.4 παρουσιάζουμε την επίδραση του μεγέθους του προβλήματος στον χρόνο επίλυσής του για, με δεμένο ότι *paths*=1.



Σχήμα 6.4: Επίδραση του μεγέθους του προβλήματος στον χρόνο επίλυσής του.

Παρατηρούμε ότι καθώς αυξάνεται ο αριθμός των διαδρομών αυξάνεται και ο χρόνος που χρειάζεται ο αλγόριθμος για να βρει την τελική βέλτιστη λύση. Ωστόσο ο αριθμός των πιλότων δεν επηρεάζει σημαντικά τον χρόνο. Όπως βλέπουμε στο σχήμα 6.4 μια μεταβολή στον αριθμό των πιλότων από 90 σε 110 με ίδιο αριθμό διαδρομών ($R=750$), αύξησε το μέσο χρόνο επίλυσης μόλις 5 λεπτά. Αντίθετα, μια μεταβολή των διαδρομών από 750 σε 550 με ίδιο αριθμό πιλότων ($N=90$), μείωσε το μέσο χρόνο επίλυσης κατά 25 λεπτά. Σημαντική είναι η αύξηση του χρόνου με την αύξηση και του αριθμού των πιλότων και του αριθμού των διαδρομών. Για αύξηση του αριθμού των πιλότων από 100 σε 150 και αύξηση του αριθμού των διαδρομών από 400 σε 900, ο χρόνος αυξήθηκε κατά 2 ώρες και 50 λεπτά. Επομένως, γίνεται αντιληπτό ότι η αύξηση των διαδρομών επηρεάζει σε μεγάλο βαθμό τον χρόνο επίλυσης, πράγμα το οποίο σε πραγματικά προβλήματα αποτελεί και μια επιπλέον δυσκολία στην εύρεση της βέλτιστης λύσης.

Όπως αναφέραμε και παραπάνω, ο χρόνος επίλυσης των προβλημάτων επηρεάζεται και σε μεγάλο βαθμό από την επιλογή της τιμής της παραμέτρου *paths*. Αυτό φαίνεται καλύτερα στο σχήμα 6.5 που παρουσιάζεται παρακάτω.



Σχήμα 6.5: Επίδραση της παραμέτρου *paths* στον χρόνο επίλυσης..

Το σχήμα 6.5 προέκυψε από τα αποτελέσματα της εφαρμογής 6 του κεφαλαίου 5. Παρατηρούμε ότι για $paths=1$ ο μέσος χρόνος επίλυσης ήταν 21 λεπτά, ενώ για $paths=2$ και για $paths=3$, οι μέσοι χρόνοι ήταν αντίστοιχα 37 λεπτά και 43 λεπτά. Βλέπουμε ότι μέχρι τώρα η διάφορα στον μέσο χρόνο επίλυσης είναι πολύ μικρή. Αξιοσημείωτη είναι η μεταβολή του μέσου χρόνου όταν επιλέγουμε να βάλουμε $paths=5$. Τότε ο χρόνος ανεβαίνει στη 1 ώρα και 19 λεπτά, δηλαδή ο μέσος χρόνος περίπου διπλασιάζεται σε σχέση με τις προηγούμενες περιπτώσεις. Τεράστια αύξηση του μέσου χρόνου έχουμε όταν επιλέγουμε $paths=10$. Σε αυτή την περίπτωση ο μέσος χρόνος ανέρχεται στις 2 ώρες και 17 λεπτά, δηλαδή 3,5 φορές πάνω από τον μέσο χρόνο για $paths=3$.

Αν ανατρέξουμε στο σχήμα 6.2 θα δούμε ότι οι μέση τιμή των λύσεων για $paths=3$, $paths=5$ και $paths=10$ είναι πολύ κοντινές, σχεδόν ίδιες. Επίσης από το σχήμα 6.3 παρατηρούμε ότι για $paths=3$ και μετά δεν έχουμε αρνητικό gap. Επομένως, συνδυάζοντας όλες αυτές τις πληροφορίες για το συγκεκριμένο πρόβλημα, καταλαβαίνουμε ότι βάζοντας παραπάνω από την τιμή 3 στην παράμετρο $paths$ δεν βελτιώνεται παραπάνω η τελική βέλτιστη λύση, αλλά το μόνο που καταφέρνουμε είναι να αυξήσουμε τον χρόνο που απαιτείται για την επίλυση του προβλήματος.

Ως απόρροια όλων των παραπάνω, μπορούμε πλέον με ασφάλεια να συμπεράνουμε πως ο αλγόριθμος λειτουργεί αποτελεσματικά σε διαφόρου μεγέθους προβλήματα βρίσκοντας τη βέλτιστη εφικτή λύση. Η χρήση του μπορεί να γίνει ακόμη και για πραγματικά προβλήματα θέτοντας τις κατάλληλες παραμέτρους και απαιτήσεις, όπως αυτές μπορούν να περιγράψουν μία αεροπορική εταιρία.

6.3 Μελλοντική Έρευνα

Ο προγραμματισμός του πληρώματος έχει υπάρξει μία από τις μεγαλύτερες επιτυχίες της επιχειρησιακής έρευνας με προγράμματα λήψης αποφάσεων να έχουν εγκατασταθεί στις μεγαλύτερες αεροπορικές εταιρίες. Ενώ οι λύσεις προ δεκαετιών

στα ημερήσια προβλήματα ήταν συνήθως 10-15% κάτω του ελάχιστου ορίου του κόστους πτήσεων, οι λύσεις πλέον βρίσκονται στο 1-2% του κατώτατου ορίου. Αυτή η βελτίωση των λύσεων μεταφράζεται σε επιπλέον κέρδη ύψους εκατομμυρίων δολαρίων ετησίως για τις μεγάλες αεροπορικές εταιρίες.

Παρόλα αυτά, ο προγραμματισμός πληρώματος στις αεροπορικές εταιρίες παραμένει ένας ενεργός κλάδος έρευνας με πολλά άλυτα προβλήματα. Μπορούν να υπάρξουν οφέλη από την ανάπτυξη καλύτερων προγραμμάτων για το πλήρωμα της καμπίνας. Αυτό το πρόβλημα έχει λάβει λιγότερη προσοχή σε σχέση με το πρόβλημα ανάθεσης πληρώματος στο πιλοτήριο κυρίως επειδή το πλήρωμα της καμπίνας στοιχίζει λιγότερο στις αεροπορικές εταιρίες και επειδή η ανάθεση του πληρώματος στο πιλοτήριο είναι ένα πολύ μεγαλύτερο πρόβλημα.

Η μεγαλύτερη πρόκληση είναι ίσως η επίλυση του προβλήματος ταιριάσματος πληρώματος, ανάθεσης πτήσεων και προγραμματισμού πτήσεων σαν ένα συνολικό πρόβλημα μιας και η επίλυση κάθε προβλήματος μεμονωμένα είναι μια πολύ δύσκολη διαδικασία.

Ο δικός μας αλγόριθμος έχει επικεντρωθεί στο πρόβλημα ανάθεσης του πληρώματος. Ωστόσο, λόγω του περιορισμένου χρόνου για τη διεκπεραίωση της μεταπτυχιακής εργασίας δεν έχει πραγματοποιηθεί κατά την επίλυση του branching, η περίπτωση όπου η τιμή της μεταβλητής που επιλέγεται την εκάστοτε φορά να παίρνει την τιμή 0. Αυτό μπορεί να αποτελέσει μια μελλοντική προσθήκη στον αλγόριθμο, η οποία ενδεχομένως θα βελτιώσει την αξιοπιστία των λύσεων. Επιπλέον, το σημαντικότερο βήμα για την αξιολόγηση του αλγορίθμου θα ήταν η εφαρμογή του σε έναν πραγματικό πρόβλημα μιας αεροπορικής εταιρίας.

BIBΛΙΟΓΡΑΦΙΑ

- Alefragis, P., Goumopoulos, C., Housos, E., Sanders, P., Takkula, T., and Wedelin, D., (1998). «*Parallel crew scheduling in PAROS*». In Proceedings of 1998 Europar/8: p. 1104–1113.
- Alefragis, P., Sanders, P., Takkula, T., and Wedelin, D., (2000). «*Parallel integer optimization for crew scheduling*». Annals of Operations Research : p. 141–166.
- Anbil, R., Gelman, E., Patty, B., and Tanga, R., (1991). «*Recent advances in crew pairing optimization at American Airlines*» : p.62–74.
- Balaji Gopalakrishnan and Ellis L. Johnson. (2005). «*Airline Crew Scheduling: State-of-the-Art*». Annals of Operations Research, vol. 140: p. 305-337.
- Campbell, K.W., R.B. Durfee, and G.S. Hines., (1997). «*FedEx Generates Bid Lines Using Simulated An-nealing*». Interfaces 27(2):p. 1–16
- Cappanera, P. and G. Gallo, (2001). «*On the Airline Crew Rostering Problem*». Technical Report TR-01-08, Department of Computer Science, University of Pisa, Italy.
- Christou, I.T., A. Zakarian, J. Liu, and H. Carter., (1999). «*A Two-Phase Genetic Algorithm for Large-Scale Bidline-Generation Problems at Delta Air Lines*». Interfaces 29(5): p. 51–65.
- Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W.P. Savelsbergh, Pamela H. Vance., (1996). «*Branch and Price: Column Generation for Solving Huge Integer Programs*». Atlanta, Georgia Institute of Technology :p. 316-329.
- Cynthia Barnhart, Amy M. Cohn, Ellis L. Johnson, Diego Klabjan, George L. Nemhauser, Pamela H. Vance., (2003). «*Handbook of Transportation Science*», Volume 56 of the series International Series in Operations Research & Management Science: p. 517-560.
- Anbil, R., Barnhart, C., Johnson, E., and Hatay, L. (1994). «*A column generation technique for the long-haul crew assignment problem*». In

Ciriani, T. and Leachman, R., editors, Optimization in Industry II, John Wiley & Sons: p. 7–24.

- Dawid, H., J. König, and C. Strauss., (2001). «*An Enhanced Rostering Model for Airline Crews*». Computers and Operations Research: p. 671–688.
- Day, P.R. and D.M. Ryan. (1997). «*Flight Attendant Rostering for Short-Haul Airline Operations*».
- Desrosiers, J., Y. Dumas, M.M. Solomon, and F. Soumis. (1995). *Time Constrained Routing and Scheduling*.». Handbooks in Operations Research and Management Science, Elsevier, Amsterdam: p. 35–139.
- Desrosiers, J., Dumas, Y., Desrochers, M., Soumis, F., Sanso, B., and Trudeau, P., (1991). «*A breakthrough in airline crew scheduling*». Technical Report G-91-11, Cahiers du GERAD.
- Diana C. Flórez, Jose L. Walteros, Miguel A. Vargas, Andrés L. Medaglia, and Nubia Velasco. (2009). «*A Mathematical Programming Approach to Airline Crew Pairing Optimization*». Department of Industrial Engineering, University of Los Andes, Bogotá, Colombia.
- Douglas Potter, (2008). «*An Implementation of a Constraint Branching Algorithm for Optimally Solving Airline Crew Pairing Problems*». Department of Mathematical Sciences, Chalmers University of Technology and Gothenburg University, Göteborg, Sweden, M.Sc. Thesis.
- F. Vanderbeck., (1994). «*Decomposition and Column Generation for Integer Programs*». PhD thesis, Universite Catholique do Lovain: p. 57-114.
- Gedron, B. and Crainic, T. (1994). «*Parallel branch-and-bound algorithms: Survey and synthesis*». Operations Research: p.1042–1066.
- George Kozanidis. (2014). «*COLUMN GENERATION FOR SCHEDULING SHIPMENTS WITHIN A SUPPLY CHAIN NETWORK WITH THE MINIMUM NUMBER OF VEHICLES*». Greece, Kos Island.
- Gershkoff, I. (1989). «*Optimizing flight crew schedules*»:p.29– 43.

- Goumopoulos, C., Housos, E., and Liljenzin, O. (1997). «*Parallel crew scheduling on workstation networks using PVM*». In Proceedings of 1997 Euro PVM-MPI.
- Jarrah, A.I.Z. and J.T. Diamond. (1997). «*The Problem of Generating Crew Bidlines*». Interfaces 27(4), p. 49–64.
- Klabjan, D. (2001). «*Next generation airline crew scheduling*». Technical report, University of Illinois at Urbana-Champaign.
- Klabjan, D. and Schwan, K. (2000). «*Airline crew pairing generation in parallel*». In Proceedings of the Tenth SIAM Conference on Parallel Processing for Scientific Computing.
- Klabjan, D., Johnson, E., Nemhauser, G., Gelman, E., and Ramaswamy, S. (2001b). «*Solving large airline crew scheduling problems: Random pairing generation and strong branching*». Computational Optimization and Applications: p.73–91.
- König, H. and C. Strauss. (2000b). «*Supplements in Airline Cabin Service*». In D. Buhalis, D.R.Fesenmaier and S. Klein (eds.), Information and Communication Technologies in Tourism 2000, p. 365–374. Berlin: Springer.
- König, H. and C. Strauss. (2000a). «*Rostering-Integrated Services and Crew Efficiency*». Information Technology and Tourism: p. 27–39.
- Lavoie, S., Minoux, M., and Odier, E. (1988). «*A new approach for crew pairing problems by column generation with an application to air transportation*». European Journal of Operational Research: p.45–58.
- Lettovsk'y, L. (1997). «*Airline Operations Recovery: An Optimization Approach*». PhD thesis, Georgia Institute of Technology.
- Lettovsk'y, L., Johnson, E., and Nemhauser, G. (2000). «*Airline crew recovery*». Transportation Science: p.337–348.
- M. Bielli, A. Bielli, R. Rossi, (2011), «*Trends in models and algorithms for fleet management*». Procedia-Social and Behavioral Sciences.
- M. E. Lübbecke and J. Desrosiers. (2002). «*Selected Topics in Column Generation*». Les Cahiers du GERAD G-2002-64, Department of

Mathematical Optimization, Braunschweig University of Technology, and GERAD: p. 66, 67, 77, 82, 91, 94, 107, 108.

- M. Grönkvist. (1998). «*Structure in Airline Crew Optimization Problems*». Master's thesis, Chalmers University of Technology, Gothenburg, Sweden: p. 22, 91, 115, 116.
- Massoud Bazargan. (2010). «*Crew Scheduling*». Airline Operations and Scheduling, 2nd ed.: p. 83-102.
- Minoux, M. (1984). «*Column generation techniques in combinatorial optimization: a new application to crew pairing problems*». In Proceedings XXIVth AGIFORS Symposium. Operations Research: p. 649–661.
- Pamela H. Vance, Cynthia Barnhart, Ellis L. Johnson, and George L. Nemhauser. (1997). «*Airline Crew Scheduling: A New Formulation and Decomposition Algorithm*». Operations Research: p. 188-200
- R. G. Sheno, (1994). «*Solving the Long Haul Crew Pairing Problem*». Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Massachusetts, M.Sc. Thesis.
- Sanders, P., Takkula, T., and Wedelin, D. (1999). «*High performance integer optimization for crew scheduling*». In Proceedings of the HPCS '99.
- Stojkovic, G., Soumis, M., and Desrosiers, J. (1998). «*The operational airline crew scheduling problem*». Transportation Science: p.232–245.
- Teodorovic, D. and Stojkovic, G. (1990). «*Model for operational daily airline scheduling*». Transportation Planning Technology: p.273–285.
- Vance, P., Barnhart, C., Johnson, E., and Nemhauser, G. (1997b). «*Airline crew scheduling: A new formulation and decomposition algorithm*». Operations Research: p.188–200.
- Wei, G. and Yu, G. (1997). «*Optimization model and algorithm for crew management during airline irregular operations*». Journal of Combinatorial Optimization: p. 305–321.

ΠΑΡΑΡΤΗΜΑ 1

Παρακάτω παρουσιάζονται η αντικειμενική συνάρτηση και οι 4 πρώτοι περιορισμοί του MP μετά από 23 επαναλήψεις του column generation.

Minimize

```
obj: 3000000 x1 + 3000000 x2 + 1000000 x3 + 1000000 x4 + 4000000 x5
      + 4000000 x6 + 1000000 x7 + 2000000 x8 + 4000000 x9 + 2000000 x10
      + 3000000 x11 + 4000000 x12 + 1000000 x13 + 2000000 x14 + 4000000 x15
      + 3000000 x16 + 1000000 x17 + 3000000 x18 + 2000000 x19 + 3000000 x20
      + 2000000 x21 + 2000000 x22 + 2000000 x23 + 2000000 x24 + 1000000 x25
      + 1000000 x26 + 2000000 x27 + 4000000 x28 + 2000000 x29 + 1000000 x30
      + 1000000 x31 + 2000000 x32 + 1000000 x33 + 4000000 x34 + 1000000 x35
      + 3000000 x36 + 2000000 x37 + 2000000 x38 + 2000000 x39 + 2000000 x40
      + 4000000 x41 + 2000000 x42 + 1000000 x43 + 4000000 x44 + 4000000 x45
      + 3000000 x46 + 3000000 x47 + 3000000 x48 + 4000000 x49 + 1000000 x50
      + 3000000 x51 + 3000000 x52 + 1000000 x53 + 1000000 x54 + 1000000 x55
      + 2000000 x56 + 2000000 x57 + 1000000 x58 + 3000000 x59 + 2000000 x60
      + 2400 x61 + 2000 x62 + 2600 x63 + 1800 x64 + 2600 x65 + 2250 x66
      + 1950 x67 + 2700 x68 + 1850 x69 + 2050 x70 + 2150 x71 + 2200 x72
      + 2050 x73 + 2000 x74 + 2100 x75 + 1950 x76 + 1950 x77 + 1900 x78
      + 2150 x79 + 2300 x80 + 1850 x81 + 1450 x82 + 11800 x83 + 1550 x84
      + 1700 x85 + 2200 x86 + 2150 x87 + 2050 x88 + 5800 x89 + 10000 x90
      + 1500 x91 + 1250 x92 + 6100 x93 + 2300 x94 + 1650 x95 + 5150 x96
      + 1950 x97 + 1300 x98 + 1500 x99 + 1650 x100 + 5950 x101 + 1750 x102
      + 1950 x103 + 7700 x104 + 1550 x105 + 4250 x106 + 1550 x107 + 1150 x108
      + 1600 x109 + 4150 x110 + 7450 x111 + 1850 x112 + 1600 x113 + 7750 x114
      + 1650 x115 + 1250 x116 + 6850 x117 + 1750 x118 + 1500 x119 + 5200 x120
      + 1250 x121 + 1350 x122 + 4600 x123 + 1350 x124 + 1450 x125 + 3750 x126
      + 1400 x127 + 1000 x128 + 5300 x129 + 1650 x130 + 1000 x131 + 1450 x132
      + 1350 x133 + 7500 x134 + 1700 x135 + 1200 x136 + 3700 x137 + 5950 x138
      + 2000 x139 + 1350 x140 + 1850 x141 + 6900 x142 + 900 x143 + 1550 x144
      + 2000 x145 + 1800 x146 + 2150 x147 + 3950 x148 + 2400 x149 + 1350 x150
      + 7650 x151 + 1250 x152 + 2150 x153 + 2300 x154 + 3850 x155 + 1800 x156
      + 1950 x157 + 3200 x158 + 1500 x159 + 1650 x160 + 4050 x161 + 1750 x162
      + 1950 x163 + 5000 x164 + 1550 x165 + 1650 x166 + 3400 x167 + 1150 x168
      + 1600 x169 + 2750 x170 + 1700 x171 + 5100 x172 + 1600 x173 + 1750 x174
      + 1650 x175 + 3950 x176 + 1600 x177 + 3050 x178 + 1500 x179 + 1800 x180
      + 1250 x181 + 5050 x182 + 4100 x183 + 1350 x184 + 1450 x185 + 1650 x186
      + 1400 x187 + 2500 x188 + 1500 x189 + 1650 x190 + 3150 x191 + 1450 x192
      + 6100 x193 + 1700 x194 + 1700 x195 + 2800 x196 + 1950 x197 + 1850 x198
      + 2000 x199 + 5050 x200 + 1850 x201 + 1450 x202 + 7000 x203 + 1550 x204
      + 1700 x205 + 1700 x206 + 2150 x207 + 2050 x208 + 2900 x209 + 1350 x210
      + 1500 x211 + 4000 x212 + 2150 x213 + 2300 x214 + 3600 x215 + 1800 x216
```

+ 1950 x217 + 1450 x218 + 1500 x219 + 1650 x220 + 3800 x221 + 1750 x222
 + 1950 x223 + 4200 x224 + 1550 x225 + 1650 x226 + 2800 x227 + 1150 x228
 + 1600 x229 + 3300 x230 + 1700 x231 + 1850 x232 + 4450 x233 + 1750 x234
 + 1650 x235 + 4900 x236 + 1600 x237 + 1750 x238 + 4450 x239 + 1800 x240
 + 1250 x241 + 3200 x242 + 3150 x243 + 1350 x244 + 1450 x245 + 1650 x246
 + 1400 x247 + 900 x248 + 1500 x249 + 1650 x250 + 3300 x251 + 1450 x252
 + 1350 x253 + 3150 x254 + 1700 x255 + 1200 x256 + 2300 x257 + 1850 x258
 + 2000 x259 + 4400 x260 + 1850 x261 + 1450 x262 + 4650 x263 + 1550 x264
 + 1700 x265 + 500 x266 + 2150 x267 + 2050 x268 + 1300 x269 + 1350 x270
 + 1500 x271 + 2800 x272 + 2150 x273 + 2300 x274 + 1900 x275 + 1800 x276
 + 1950 x277 + 1900 x278 + 1500 x279 + 550 x280 + 1750 x281 + 1750 x282
 + 1950 x283 + 2300 x284 + 1550 x285 + 1650 x286 + 1850 x287 + 1000 x288
 + 1600 x289 + 700 x290 + 1700 x291 + 1850 x292 + 4400 x293 + 1750 x294
 + 1650 x295 + 4200 x296 + 1600 x297 + 1750 x298 + 2500 x299 + 1800 x300
 + 1250 x301 + 2700 x302 + 1900 x303 + 1350 x304 + 3400 x305 + 1650 x306
 + 1400 x307 + 1050 x308 + 1500 x309 + 1650 x310 + 3300 x311 + 1450 x312
 + 1350 x313 + 2500 x314 + 1700 x315 + 1200 x316 + 1700 x317 + 1850 x318
 + 2000 x319 + 3100 x320 + 1850 x321 + 1450 x322 + 3200 x323 + 1550 x324
 + 1700 x325 + 350 x326 + 2150 x327 + 2050 x328 + 2850 x329 + 1350 x330
 + 1500 x331 + 3000 x332 + 2150 x333 + 2300 x334 + 1800 x336 + 1950 x337
 + 1650 x338 + 1500 x339 + 1800 x340 + 1750 x341 + 1750 x342 + 1950 x343
 + 1300 x344 + 1550 x345 + 1650 x346 + 850 x347 + 1150 x348 + 1600 x349
 + 1450 x350 + 1700 x351 + 1850 x352 + 2650 x353 + 1750 x354 + 1650 x355
 + 1900 x356 + 1600 x357 + 1750 x358 + 1000 x359 + 1800 x360 + 1250 x361
 + 2800 x362 + 1900 x363 + 1350 x364 + 1700 x365 + 1650 x366 + 1400 x367
 + 950 x368 + 1500 x369 + 1650 x370 + 1850 x371 + 1450 x372 + 1350 x373
 + 1700 x374 + 1700 x375 + 1200 x376 + 1350 x377 + 1850 x378 + 2000 x379
 + 2450 x380 + 1850 x381 + 1450 x382 + 4200 x383 + 1550 x384 + 1700 x385
 + 2150 x387 + 2050 x388 + 300 x389 + 1350 x390 + 1500 x391 + 1750 x392
 + 2150 x393 + 2300 x394 + 1400 x395 + 1800 x396 + 1950 x397 + 600 x398
 + 1500 x399 + 1350 x400 + 1750 x401 + 1750 x402 + 1950 x403 + 1300 x404
 + 1550 x405 + 1650 x406 + 2250 x407 + 1150 x408 + 1600 x409 + 1450 x410
 + 1700 x411 + 1850 x412 + 2150 x413 + 1750 x414 + 1650 x415 + 2250 x416
 + 1600 x417 + 1750 x418 + 3650 x419 + 1800 x420 + 1250 x421 + 2850 x422
 + 1900 x423 + 1350 x424 + 200 x425 + 1650 x426 + 1400 x427 + 50 x428
 + 1500 x429 + 1650 x430 + 900 x431 + 1450 x432 + 1350 x433 + 1600 x434
 + 1700 x435 + 1200 x436 + 650 x437 + 1850 x438 + 2000 x439 + 2850 x440
 + 1850 x441 + 1450 x442 + 2400 x443 + 1550 x444 + 1700 x445 + 600 x446
 + 2150 x447 + 2050 x448 + 950 x449 + 1350 x450 + 1500 x451 + 1750 x452

+ 2150 x453 + 2300 x454 + 650 x455 + 1800 x456 + 1950 x457 + 550 x458
 + 1500 x459 + 1650 x460 + 3650 x461 + 1750 x462 + 1950 x463 + 700 x464
 + 1550 x465 + 1650 x466 + 1350 x467 + 1150 x468 + 1600 x469 + 700 x470
 + 1700 x471 + 1850 x472 + 2900 x473 + 1750 x474 + 1650 x475 + 3900 x476
 + 1600 x477 + 1750 x478 + 1600 x479 + 1800 x480 + 1250 x481 + 850 x482
 + 1900 x483 + 1350 x484 + 2200 x485 + 1650 x486 + 1400 x487 + 1000 x488
 + 1500 x489 + 1650 x490 + 1400 x491 + 1450 x492 + 1350 x493 + 1950 x494
 + 1700 x495 + 1200 x496 + 1650 x497 + 1850 x498 + 2000 x499 + 800 x500
 + 1850 x501 + 1450 x502 + 5600 x503 + 1550 x504 + 1700 x505 + 2150 x507
 + 2050 x508 + 1650 x509 + 1350 x510 + 1500 x511 + 4900 x512 + 2150 x513
 + 2300 x514 + 2250 x515 + 1800 x516 + 1950 x517 + 1400 x518 + 1500 x519
 + 1650 x520 + 2450 x521 + 1750 x522 + 1950 x523 + 2450 x524 + 1550 x525
 + 1650 x526 + 1500 x527 + 1150 x528 + 1600 x529 + 2650 x530 + 1700 x531
 + 1850 x532 + 3000 x533 + 1750 x534 + 1650 x535 + 4950 x536 + 1600 x537
 + 1750 x538 + 1050 x539 + 1800 x540 + 1250 x541 + 2500 x542 + 1900 x543
 + 1350 x544 + 2650 x545 + 1650 x546 + 1400 x547 + 1450 x548 + 1500 x549
 + 1650 x550 + 2100 x551 + 1450 x552 + 1350 x553 + 2050 x554 + 1700 x555
 + 1200 x556 + 1600 x557 + 1850 x558 + 2000 x559 + 1150 x560 + 1850 x561
 + 1450 x562 + 50 x563 + 1550 x564 + 1700 x565 + 1000 x566 + 2150 x567
 + 2050 x568 + 650 x569 + 1350 x570 + 1500 x571 + 900 x572 + 2150 x573
 + 2300 x574 + 400 x575 + 1800 x576 + 1950 x577 + 1350 x578 + 1500 x579
 + 1650 x580 + 100 x581 + 1750 x582 + 1950 x583 + 250 x584 + 1550 x585
 + 1650 x586 + 1150 x588 + 1600 x589 + 1300 x590 + 1700 x591 + 1850 x592
 + 500 x593 + 1750 x594 + 1650 x595 + 900 x596 + 1600 x597 + 1750 x598
 + 400 x599 + 1800 x600 + 600 x601 + 1350 x602 + 1900 x603 + 650 x604
 + 1450 x605 + 1650 x606 + 1400 x607 + 1500 x609 + 1650 x610 + 300 x611
 + 1450 x612 + 1350 x613 + 1700 x615 + 1200 x616 + 600 x617 + 1850 x618
 + 2000 x619 + 150 x620 + 1850 x621 + 1450 x622 + 3000 x623 + 1550 x624
 + 1700 x625 + 2150 x627 + 2050 x628 + 350 x629 + 1350 x630 + 1500 x631
 + 1600 x632 + 2150 x633 + 2300 x634 + 1400 x635 + 1800 x636 + 1950 x637
 + 200 x638 + 1500 x639 + 1650 x640 + 850 x641 + 1750 x642 + 1950 x643
 + 1050 x644 + 1550 x645 + 1650 x646 + 150 x647 + 1150 x648 + 1600 x649
 + 50 x650 + 1700 x651 + 1850 x652 + 1000 x653 + 1750 x654 + 1650 x655
 + 2050 x656 + 1600 x657 + 1750 x658 + 350 x659 + 1800 x660 + 1250 x661
 + 2000 x662 + 1900 x663 + 1350 x664 + 150 x665 + 1650 x666 + 1400 x667
 + 900 x668 + 1500 x669 + 1650 x670 + 1700 x671 + 1450 x672 + 1350 x673
 + 900 x674 + 1700 x675 + 1200 x676 + 100 x677 + 1850 x678 + 2000 x679
 + 300 x680 + 1850 x681 + 1450 x682 + 850 x683 + 1550 x684 + 1700 x685
 + 50 x686 + 2150 x687 + 2050 x688 + 50 x689 + 1350 x690 + 1500 x691

+ 1650 x692 + 2150 x693 + 2300 x694 + 900 x695 + 1800 x696 + 1950 x697
 + 150 x698 + 1500 x699 + 1650 x700 + 1000 x701 + 1750 x702 + 1950 x703
 + 750 x704 + 1550 x705 + 1650 x706 + 150 x707 + 1150 x708 + 1600 x709
 + 450 x710 + 1700 x711 + 1850 x712 + 1000 x713 + 1750 x714 + 1650 x715
 + 500 x716 + 1600 x717 + 1750 x718 + 400 x719 + 1800 x720 + 1250 x721
 + 1900 x723 + 1350 x724 + 950 x725 + 1650 x726 + 1400 x727 + 1450 x728
 + 1500 x729 + 1650 x730 + 1700 x731 + 1450 x732 + 1350 x733 + 1400 x734
 + 1700 x735 + 1200 x736 + 100 x737 + 1850 x738 + 2000 x739 + 400 x740
 + 1850 x741 + 1450 x742 + 1500 x743 + 1550 x744 + 1700 x745 + 300 x746
 + 2150 x747 + 2050 x748 + 1150 x749 + 1350 x750 + 1500 x751 + 300 x752
 + 2150 x753 + 2300 x754 + 1800 x756 + 1950 x757 + 600 x758 + 1500 x759
 + 1650 x760 + 600 x761 + 1750 x762 + 1900 x763 + 1000 x764 + 1550 x765
 + 1650 x766 + 50 x767 + 1150 x768 + 1600 x769 + 1150 x770 + 1700 x771
 + 1850 x772 + 2000 x773 + 1750 x774 + 1650 x775 + 300 x776 + 1600 x777
 + 1750 x778 + 450 x779 + 1800 x780 + 1250 x781 + 1900 x783 + 1350 x784
 + 1150 x785 + 1650 x786 + 1400 x787 + 1100 x788 + 1500 x789 + 1650 x790
 + 1450 x792 + 1350 x793 + 1400 x794 + 1700 x795 + 1200 x796 + 1850 x798
 + 2000 x799 + 1150 x800 + 1850 x801 + 1450 x802 + 2500 x803 + 1550 x804
 + 1700 x805 + 50 x806 + 2150 x807 + 2050 x808 + 1450 x809 + 1350 x810
 + 1500 x811 + 1950 x812 + 2150 x813 + 2300 x814 + 150 x815 + 1800 x816
 + 1950 x817 + 1500 x819 + 1650 x820 + 800 x821 + 1750 x822 + 1600 x823
 + 1000 x824 + 1550 x825 + 1650 x826 + 850 x827 + 1150 x828 + 1600 x829
 + 350 x830 + 1700 x831 + 1850 x832 + 3000 x833 + 1750 x834 + 1650 x835
 + 1250 x836 + 1600 x837 + 1750 x838 + 850 x839 + 1800 x840 + 1250 x841
 + 1900 x843 + 1350 x844 + 1850 x845 + 1650 x846 + 1400 x847 + 1200 x848
 + 1500 x849 + 1650 x850 + 200 x851 + 1450 x852 + 1350 x853 + 1800 x854
 + 1700 x855 + 1200 x856 + 1250 x857 + 1850 x858 + 2000 x859 + 100 x860
 + 1850 x861 + 1450 x862 + 400 x863 + 1550 x864 + 1700 x865 + 350 x866
 + 2150 x867 + 2050 x868 + 1350 x870 + 1500 x871 + 2150 x873 + 2300 x874
 + 550 x875 + 1800 x876 + 1950 x877 + 500 x878 + 1500 x879 + 1650 x880
 + 150 x881 + 1750 x882 + 1000 x884 + 1550 x885 + 1650 x886 + 1150 x888
 + 1600 x889 + 1700 x891 + 1850 x892 + 1750 x894 + 1650 x895 + 100 x896
 + 1600 x897 + 1750 x898 + 50 x899 + 1800 x900 + 1250 x901 + 1900 x903
 + 1350 x904 + 1650 x906 + 1400 x907 + 1500 x909 + 1650 x910 + 1450 x912
 + 1350 x913 + 450 x914 + 1700 x915 + 1200 x916 + 350 x917 + 1850 x918
 + 2000 x919 + 1850 x921 + 1450 x922 + 1550 x924 + 1700 x925 + 50 x926
 + 2150 x927 + 2050 x928 + 50 x929 + 1350 x930 + 1500 x931 + 2150 x933
 + 2300 x934 + 50 x935 + 1800 x936 + 1950 x937 + 1500 x939 + 1650 x940
 + 300 x941 + 1750 x942 + 1950 x943 + 300 x944 + 1550 x945 + 1650 x946

+ 450 x947 + 1150 x948 + 1600 x949 + 150 x950 + 1700 x951 + 1850 x952
 + 100 x953 + 1750 x954 + 1650 x955 + 400 x956 + 1600 x957 + 1750 x958
 + 1800 x960 + 1250 x961 + 1900 x963 + 1350 x964 + 50 x965 + 1650 x966
 + 1400 x967 + 1500 x969 + 1650 x970 + 50 x971 + 1450 x972 + 1350 x973
 + 1700 x975 + 1200 x976 + 1850 x978 + 2000 x979 + 250 x980 + 1850 x981
 + 1450 x982 + 1550 x984 + 1700 x985 + 2150 x987 + 2050 x988 + 1350 x990
 + 1500 x991 + 2150 x993 + 2300 x994 + 1800 x996 + 1950 x997 + 50 x998
 + 1500 x999 + 1650 x1000 + 1750 x1002 + 1950 x1003 + 1550 x1005
 + 100 x1006 + 1550 x1007 + 1150 x1008 + 1600 x1009 + 300 x1010
 + 1700 x1011 + 1850 x1012 + 50 x1013 + 1750 x1014 + 1650 x1015
 + 100 x1016 + 1600 x1017 + 1750 x1018 + 1800 x1020 + 1250 x1021
 + 1900 x1023 + 1350 x1024 + 400 x1025 + 1650 x1026 + 1400 x1027
 + 150 x1028 + 1500 x1029 + 1650 x1030 + 1450 x1032 + 1350 x1033
 + 200 x1034 + 1700 x1035 + 1200 x1036 + 1850 x1038 + 2000 x1039
 + 150 x1040 + 1850 x1041 + 1450 x1042 + 1550 x1044 + 1700 x1045
 + 2150 x1047 + 2050 x1048 + 1350 x1050 + 1500 x1051 + 2150 x1053
 + 2300 x1054 + 100 x1055 + 1800 x1056 + 1950 x1057 + 1500 x1059
 + 1650 x1060 + 1950 x1063 + 1000 x1064 + 1550 x1065 + 1650 x1066
 + 50 x1067 + 1150 x1068 + 700 x1070 + 1700 x1071 + 1850 x1072
 + 1750 x1074 + 1650 x1075 + 1600 x1077 + 1750 x1078 + 1800 x1080
 + 1250 x1081 + 1900 x1083 + 1350 x1084 + 1650 x1086 + 1400 x1087
 + 100 x1088 + 1500 x1089 + 1650 x1090 + 50 x1091 + 1450 x1092
 + 1350 x1093 + 1700 x1095 + 1200 x1096 + 50 x1097 + 1850 x1098
 + 2000 x1099 + 50 x1100 + 1850 x1101 + 1450 x1102 + 1550 x1104
 + 1700 x1105 + 50 x1106 + 2150 x1107 + 2050 x1108 + 1350 x1110
 + 1500 x1111 + 2150 x1113 + 2300 x1114 + 1800 x1116 + 1950 x1117
 + 1500 x1119 + 1650 x1120 + 200 x1121 + 1750 x1122 + 1950 x1123
 + 50 x1124 + 1550 x1125 + 1550 x1127 + 1150 x1128 + 1600 x1129
 + 1700 x1131 + 1850 x1132 + 100 x1133 + 1750 x1134 + 1650 x1135
 + 1600 x1137 + 1750 x1138 + 1800 x1140 + 1250 x1141 + 100 x1142
 + 1900 x1143 + 1350 x1144 + 1650 x1146 + 1400 x1147 + 150 x1148
 + 1500 x1149 + 1650 x1150 + 100 x1151 + 1450 x1152 + 1350 x1153
 + 1700 x1155 + 1200 x1156 + 100 x1157 + 1850 x1158 + 2000 x1159
 + 150 x1160 + 1850 x1161 + 1450 x1162 + 200 x1163 + 1550 x1164
 + 1700 x1165 + 50 x1166 + 2150 x1167 + 2050 x1168 + 150 x1169
 + 1350 x1170 + 1500 x1171 + 2150 x1173 + 2300 x1174 + 150 x1175
 + 1800 x1176 + 1950 x1177 + 100 x1178 + 1500 x1179 + 50 x1180
 + 1750 x1181 + 1750 x1182 + 1950 x1183 + 1550 x1185 + 1650 x1186
 + 50 x1187 + 1150 x1188 + 1600 x1189 + 1700 x1191 + 1850 x1192 + 50 x1193

+ 1750 x1194 + 1650 x1195 + 1600 x1197 + 1750 x1198 + 1800 x1200
 + 1250 x1201 + 100 x1202 + 1900 x1203 + 1350 x1204 + 50 x1205
 + 1650 x1206 + 1400 x1207 + 1500 x1209 + 1650 x1210 + 1450 x1212
 + 1350 x1213 + 1700 x1215 + 1200 x1216 + 1850 x1218 + 2000 x1219
 + 100 x1220 + 1850 x1221 + 1450 x1222 + 1550 x1224 + 1800 x1226
 + 2150 x1227 + 2400 x1229 + 1350 x1230 + 1500 x1231 + 2150 x1233
 + 2300 x1234 + 1800 x1236 + 1950 x1237 + 1500 x1239 + 1650 x1240
 + 1750 x1242 + 1950 x1243 + 1550 x1245 + 1650 x1246 + 50 x1248
 + 1600 x1249 + 700 x1250 + 1700 x1251 + 1850 x1252 + 250 x1253
 + 1750 x1254 + 1650 x1255 + 1600 x1257 + 1750 x1258 + 1800 x1260
 + 1250 x1261 + 1900 x1263 + 1350 x1264 + 50 x1265 + 1650 x1266
 + 1400 x1267 + 50 x1268 + 1500 x1269 + 1650 x1270 + 50 x1271 + 1450 x1272
 + 1700 x1274 + 1700 x1275 + 1950 x1277 + 1850 x1278 + 2000 x1279
 + 1850 x1281 + 1450 x1282 + 1550 x1284 + 1700 x1285 + 2150 x1287
 + 2050 x1288 + 50 x1289 + 1350 x1290 + 1500 x1291 + 2150 x1293
 + 2300 x1294 + 1800 x1296 + 1950 x1297 + 150 x1298 + 1500 x1299
 + 1650 x1300 + 50 x1301 + 1750 x1302 + 1950 x1303 + 300 x1304
 + 1550 x1305 + 1650 x1306 + 250 x1307 + 1150 x1308 + 1600 x1309
 + 1700 x1311 + 1850 x1312 + 1750 x1314 + 1650 x1315 + 1600 x1317
 + 1750 x1318 + 250 x1319 + 1800 x1320 + 1250 x1321 + 200 x1322
 + 1900 x1323 + 1350 x1324 + 1650 x1326 + 1400 x1327 + 250 x1328
 + 1500 x1329 + 1650 x1330 + 300 x1331 + 1450 x1332 + 1350 x1333
 + 1700 x1335 + 1200 x1336 + 500 x1337 + 1850 x1338 + 2000 x1339
 + 1850 x1341 + 1450 x1342 + 1550 x1344 + 1700 x1345 + 150 x1346
 + 2150 x1347 + 2050 x1348 + 150 x1349 + 1350 x1350 + 1500 x1351
 + 300 x1352 + 2150 x1353 + 2300 x1354 + 50 x1355 + 1800 x1356
 + 1950 x1357 + 1500 x1359 + 1650 x1360 + 250 x1361 + 1750 x1362
 + 1950 x1363 + 50 x1364 + 1550 x1365 + 1650 x1366 + 1150 x1368
 + 1600 x1369 + 1700 x1371 + 1850 x1372 + 100 x1373 + 1750 x1374
 + 1650 x1375 + 100 x1376 + 1600 x1377 + 1750 x1378 + 50 x1379
 + 1800 x1380 + 1250 x1381 + 300 x1382 + 1900 x1383 + 1350 x1384
 + 200 x1385 + 1650 x1386 + 1400 x1387 + 50 x1388 + 1500 x1389
 + 1650 x1390 + 1450 x1392 + 1350 x1393 + 200 x1394 + 1700 x1395
 + 1200 x1396 + 250 x1397 + 1850 x1398 + 2000 x1399 + 300 x1400
 + 1850 x1401 + 1450 x1402 + 1550 x1404 + 1700 x1405 + 2150 x1407
 + 2050 x1408 + 50 x1409 + 1350 x1410 + 1500 x1411 + 100 x1412
 + 2150 x1413 + 2300 x1414 + 250 x1415 + 1800 x1416 + 1950 x1417
 + 50 x1418 + 1500 x1419 + 1650 x1420 + 1950 x1423 + 1000 x1424
 + 1550 x1425 + 1550 x1427 + 1150 x1428 + 1600 x1429 + 400 x1430

+ 1700 x1431 + 1850 x1432 + 50 x1433 + 1750 x1434 + 1650 x1435 + 50 x1436
 + 1600 x1437 + 1750 x1438 + 50 x1439 + 1800 x1440 + 1250 x1441
 + 150 x1442 + 1900 x1443 + 1350 x1444 + 1650 x1446 + 1400 x1447
 + 50 x1448 + 1500 x1449 + 1650 x1450 + 150 x1451 + 1450 x1452
 + 1350 x1453 + 1700 x1455 + 1200 x1456 + 250 x1457 + 1850 x1458
 + 2000 x1459

Subject To

- c1: x61 + x81 + x82 + x83 + x141 + x142 + x143 + x201 + x202 + x203 + x261
 + x262 + x263 + x321 + x322 + x323 + x381 + x382 + x383 + x441 + x442
 + x443 + x501 + x502 + x503 + x561 + x562 + x563 + x621 + x622 + x623
 + x681 + x682 + x683 + x741 + x742 + x743 + x801 + x802 + x803 + x861
 + x862 + x863 + x921 + x922 + x923 + x981 + x982 + x983 + x1041 + x1042
 + x1043 + x1101 + x1102 + x1103 + x1161 + x1162 + x1163 + x1221 + x1222
 + x1223 + x1281 + x1282 + x1283 + x1341 + x1342 + x1343 + x1401 + x1402
 + x1403 = 1
- c2: x62 + x84 + x85 + x86 + x144 + x145 + x146 + x204 + x205 + x206 + x264
 + x265 + x266 + x324 + x325 + x326 + x384 + x385 + x386 + x444 + x445
 + x446 + x504 + x505 + x506 + x564 + x565 + x566 + x624 + x625 + x626
 + x684 + x685 + x686 + x744 + x745 + x746 + x804 + x805 + x806 + x864
 + x865 + x866 + x924 + x925 + x926 + x984 + x985 + x986 + x1044 + x1045
 + x1046 + x1104 + x1105 + x1106 + x1164 + x1165 + x1166 + x1224 + x1225
 + x1226 + x1284 + x1285 + x1286 + x1344 + x1345 + x1346 + x1404 + x1405
 + x1406 = 1
- c3: x63 + x87 + x88 + x89 + x147 + x148 + x149 + x207 + x208 + x209 + x267
 + x268 + x269 + x327 + x328 + x329 + x387 + x388 + x389 + x447 + x448
 + x449 + x507 + x508 + x509 + x567 + x568 + x569 + x627 + x628 + x629
 + x687 + x688 + x689 + x747 + x748 + x749 + x807 + x808 + x809 + x867
 + x868 + x869 + x927 + x928 + x929 + x987 + x988 + x989 + x1047 + x1048
 + x1049 + x1107 + x1108 + x1109 + x1167 + x1168 + x1169 + x1227 + x1228
 + x1229 + x1287 + x1288 + x1289 + x1347 + x1348 + x1349 + x1407 + x1408
 + x1409 = 1
- c4: x64 + x90 + x91 + x92 + x150 + x151 + x152 + x210 + x211 + x212 + x270
 + x271 + x272 + x330 + x331 + x332 + x390 + x391 + x392 + x450 + x451
 + x452 + x510 + x511 + x512 + x570 + x571 + x572 + x630 + x631 + x632
 + x690 + x691 + x692 + x750 + x751 + x752 + x810 + x811 + x812 + x870
 + x871 + x872 + x930 + x931 + x932 + x990 + x991 + x992 + x1050 + x1051
 + x1052 + x1110 + x1111 + x1112 + x1170 + x1171 + x1172 + x1230 + x1231
 + x1232 + x1290 + x1291 + x1292 + x1350 + x1351 + x1352 + x1410 + x1411
 + x1412 = 1

ΠΑΡΑΡΤΗΜΑ 2

N=90 R=550 paths=1						
ΠΡΟΒΛΗΜΑ	Time	Λύση MP	Τελική βέλτιστη λύση	gap	Uncovered	Branching
1	0:18:33	50766	50800	34	0	89
2	0:21:34	49129	49100	-29	0	89
3	0:14:21	58015	58000	-15	0	89
4	0:16:51	51943	51950	7	0	89
5	0:26:18	44794	45000	206	0	89
6	0:30:38	43343	43450	107	0	89
7	0:28:50	45869	45950	81	0	89
8	0:23:57	41675	2041100	1999425	1	89
9	0:20:07	50505	51000	495	0	89
10	0:23:40	46341	46400	59	0	89
11	0:23:10	56843	56800	-43	0	89
12	0:24:29	51778	51850	72	0	89
13	0:21:19	47107	46850	-257	0	88
14	0:17:19	42800	42850	50	0	89
15	0:16:38	54234	54150	-84	0	88
16	0:19:35	41252	41250	-2	0	89
17	0:16:52	48949	48900	-49	0	89

18	0:21:27	45692	46000	308	0	89
19	0:15:38	38919	38850	-69	0	88
20	0:17:39	42260	42400	140	0	88